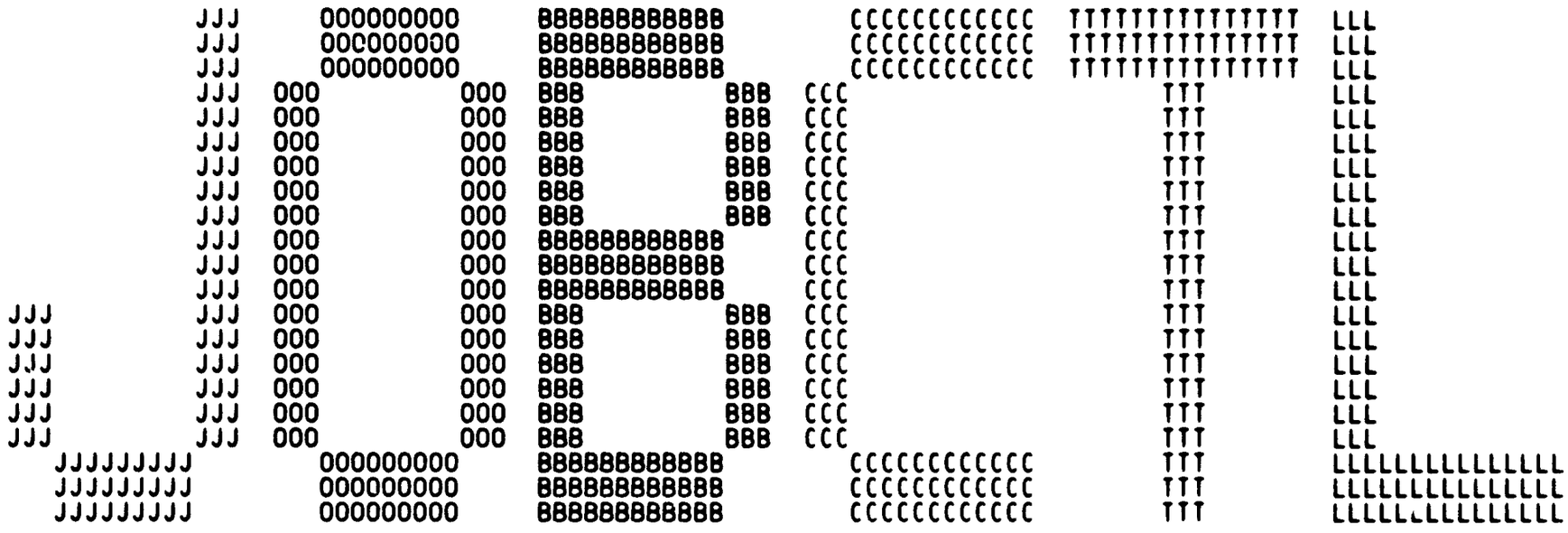


_$2

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

```
  AAAAAA      SSSSSSSS  YY      YY  NN      NN    CCCCCCCC  HH      HH  RRRRRRRR      000000    NN      NN
  AAAAAA      SSSSSSSS  YY      YY  NN      NN    CCCCCCCC  HH      HH  RRRRRRRR      000000    NN      NN
AA      AA  SS          YY      YY  NN      NN  CC          HH      HH  RR      RR  00      00  NN      NN
AA      AA  SS          YY      YY  NN      NN  CC          HH      HH  RR      RR  00      00  NN      NN
AA      AA  SS            YY  YY    NNNN    NN  CC          HH      HH  RR      RR  00      00  NNNN    NN
AA      AA  SS            YY  YY    NNNN    NN  CC          HH      HH  RR      RR  00      00  NNNN    NN
AA      AA    SSSSSS        YY      NN  NN  NN  CC          HHHHHHHHHH  RRRRRRR     00      00  NN  NN  NN
AA      AA    SSSSSS        YY      NN  NN  NN  CC          HHHHHHHHHH  RRRRRRR     00      00  NN  NN  NN
AAAAAAAAAA          SS      YY      NN    NNNN  CC          HH      HH  RR  RR      00      00  NN    NNNN
AAAAAAAAAA          SS      YY      NN    NNNN  CC          HH      HH  RR  RR      00      00  NN    NNNN
AA      AA          SS      YY      NN      NN  CC          HH      HH  RR    RR    00      00  NN      NN      ....
AA      AA          SS      YY      NN      NN  CC          HH      HH  RR    RR    00      00  NN      NN      ....
AA      AA  SSSSSSSS        YY      NN      NN    CCCCCCCC  HH      HH  RR      RR    000000    NN      NN      ....
AA      AA  SSSSSSSS        YY      NN      NN    CCCCCCCC  HH      HH  RR      RR    000000    NN      NN      ....

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 ASYNCHRON(%TITLE 'Asynchronous service management'
;     2          0002  0                    IDENT = 'V04-002'
;     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 THFREOF 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 that manage services that complete
;    37          0037  1 !       asynchronously to the original request.  Many such instances require
;    38          0038  1 !       communication with remote job controllers in a cluster.
;    39          0039  1 !
;    40          0040  1 ! ENVIRONMENT:
;    41          0041  1 !       VAX/VMS user and kernel mode.
;    42          0042  1 !--
;    43          0043  1 !
;    44          0044  1 ! AUTHOR: M. Jack, CREATION DATE: 16-Feb-1982
;    45          0045  1 !
;    46          0046  1 ! MODIFIED BY:
;    47          0047  1 !
;    48          0048  1 !       V04-002 JAK0236         J A Krycka      14-Sep-1984
;    49          0049  1 !               Collect more diagnostic information.
;    50          0050  1 !
;    51          0051  1 !       V04-001 JAK0235         J A Krycka      12-Sep-1984
;    52          0052  1 !               Detect and repair a corrupted incomplete services list in
;    53          0053  1 !               SCAN_INCOMPLETE_SERVICES.
;    54          0054  1 !
;    55          0055  1 !       V03-011 JAK0224         J A Krycka      24-Aug-1984
;    56          0056  1 !               In ENTER_REMOTE_REQUEST set a flag if there is no doorbell lock
;    57          0057  1 !               defined for the remote job controller (indicating that the
```

```
;    58          0058  1 !               remote node is not in the cluster or the remote job controller
;    59          0059  1 !               does not have the queue file open).
;    60          0060  1 !
;    61          0061  1 !       V03-010 KPL0003         P Lieberwirth,  30-Jul-1984
;    62          0062  1 !               Fix ALL bugs introduced in V03-008.
;    63          0063  1 !
;    64          0064  1 !       V03-009 KPL0002         P Lieberwirth,  30-Jul-1984
;    65          0065  1 !               Sigh, fix bug in V03-008.  When rewriting predecessor,
;    66          0066  1 !               do not update predecessor pointer to be the deallocated
;    67          0067  1 !               record.  (That took someone with the brain of a turnip.)
;    68          0068  1 !
;    69          0069  1 !       V03-008 KPL0001         P Lieberwirth,  19-Jul-1984
;    70          0070  1 !               Rewrite predecessor before deallocating SRQ in routine
;    71          0071  1 !               SCAN_INCOMPLETE_SERVICES.  This avoids corrupting the
;    72          0072  1 !               incomplete service list at the possible cost of losing
;    73          0073  1 !               a deallocated record and extra IOs.
;    74          0074  1 !
;    75          0075  1 !       V03-007 JAK0213         J A Krycka      18-May-1984
;    76          0076  1 !               Continuation of V03-006. Use newly created LCK$M_NODLCKBLK
;    77          0077  1 !               (no deadlock on blocking AST) option on enqueue instead.
;    78          0078  1 !
;    79          0079  1 !       V03-006 JAK0208         J A Krycka      08-May-1984
;    80          0080  1 !               Use LCK$M_NODLCKWT (no deadlock wait) option on enqueue service
;    81          0081  1 !               for the remote doorbell lock to avoid having the lock manager
;    82          0082  1 !               declare a deadlock situation.
;    83          0083  1 !
;    84          0084  1 !       V03-005 GRR0001         Gregory R. Robert       09-Sep-1983
;    85          0085  1 !               Supply missing dot in call to delete_files.
;    86          0086  1 !
;    87          0087  1 !       V03-004 MLJ0115         Martin L. Jack, 30-Jul-1983
;    88          0088  1 !               Changes for job controller baselevel.
;    89          0089  1 !
;    90          0090  1 !       V03-003 MLJ0114         Martin L. Jack, 23-Jun-1983
;    91          0091  1 !               Changes for job controller baselevel.
;    92          0092  1 !
;    93          0093  1 !       V03-002 MLJ0113         Martin L. Jack, 26-May-1983
;    94          0094  1 !               Changes for job controller baselevel.
;    95          0095  1 !
;    96          0096  1 !       V03-001 MLJ0112         Martin L. Jack, 29-Apr-1983
;    97          0097  1 !               Changes for job controller baselevel.
;    98          0098  1 !
;    99          0099  1 !**
```

```
;  101     0100  1 REQUIRE 'SRC$:JOBCTLDEF';
;  102     1141  1
;  103     1142  1
;  104     1143  1 FORWARD ROUTINE
;  105     1144  1         CREATE_SRQ_RECORD,
;  106     1145  1         PROCESS_REMOTE_SERVICES:        L_OUTPUT_1,
;  107     1146  1         SCAN_INCOMPLETE_SERVICES:       NOVALUE,
;  108     1147  1         REMOTE_BLOCKING_AST:            NOVALUE,
;  109     1148  1         REMOTE_COMPLETION_NONAST:       NOVALUE,
;  110     1149  1         REMOTE_COMPLETION_AST:          NOVALUE,
;  111     1150  1         ENTER_REMOTE_REQUEST:           NOVALUE,
;  112     1151  1         ENTER_REMOTE_REQUEST_AST:       NOVALUE,
;  113     1152  1         QUEUE_MASTER_AST:               NOVALUE;
;  114     1153  1
;  115     1154  1
;  116     1155  1 EXTERNAL ROUTINE
;  117     1156  1         ABORT_EXECUTION,
;  118     1157  1         AFTER_AST:                      NOVALUE,
;  119     1158  1         ALLOCATE_MEMORY,
;  120     1159  1         ALLOCATE_RECORD:                L_OUTPUT_2,
;  121     1160  1         BROADCAST_MESSAGE:              NOVALUE,
;  122     1161  1         COMPLETE_JOB:                   NOVALUE,
;  123     1162  1         CREATE_SRB:                     NOVALUE,
;  124     1163  1         DEALLOCATE_MEMORY:              NOVALUE,
;  125     1164  1         DEALLOCATE_RECORD:              NOVALUE,
;  126     1165  1         DELETE_FILES:                   NOVALUE,
;  127     1166  1         FIND_PENDING_JOBS:              NOVALUE,
;  128     1167  1         FLUSH_RECORD:                   NOVALUE,
;  129     1168  1         LOCK_QUEUE_FILE.                NOVALUE,
;  130     1169  1         PAUSE_EXECUTION,
;  131     1170  1         READ_RECORD,
;  132     1171  1         RELEASE_RECORD:                 NOVALUE,
;  133     1172  1         RESET_EXECUTOR_QUEUE:           NOVALUE,
;  134     1173  1         RESUME_EXECUTION,
;  135     1174  1         REWRITE_RECORD:                 NOVALUE,
;  136     1175  1         SCHEDULE_NONAST:                NOVALUE,
;  137     1176  1         SEND_SERVICE_RESPONSE_MESSAGE:  NOVALUE,
;  138     1177  1         START_EXECUTION,
;  139     1178  1         START_SYMBIONT_STREAM,
;  140     1179  1         STOP_SYMBIONT_STREAM,
;  141     1180  1         UNLOCK_QUEUE_FILE:              NOVALUE,
;  142     1181  1         UPDATE_GETQUI_DATA:             NOVALUE;
;  143     1182  1
;  144     1183  1
;  145     1184  1 LITERAL
;  146     1185  1         K_COMPLETE=     0,              ! Complete request with status
;  147     1186  1         K_DEALLOCATE=   1,              ! Deallocate request
;  148     1187  1         K_RELEASE=      2,              ! Leave request in queue
;  149     1188  1         K_REWRITE=      3;              ! Leave request in queue and rewrite
;  150     1189  1
;  151     1190  1
;  152     1191  1 BUILTIN
;  153     1192  1         TESTBITSC,
;  154     1193  1         TESTBITSS;
```

```
;  156          1194  1 GLOBAL ROUTINE CREATE_SRQ_RECORD(FUNC,P1,P2,P3,P4,P5,P6,P7)=
;  157          1195  1
;  158          1196  1 !++
;  159          1197  1 !
;  160          1198  1 ! FUNCTIONAL DESCRIPTION:
;  161          1199  1 !       This routine allocates, initializes, and enqueues an incomplete service
;  162          1200  1 !       record.
;  163          1201  1 !
;  164          1202  1 ! INPUT PARAMETERS:
;  165          1203  1 !       FUNC            - Function code.
;  166          1204  1 !       P1-P7           - Function-specific parameters.
;  167          1205  1 !
;  168          1206  1 ! IMPLICIT INPUTS:
;  169          1207  1 !       NONE
;  170          1208  1 !
;  171          1209  1 ! OUTPUT PARAMETERS:
;  172          1210  1 !       NONE
;  173          1211  1 !
;  174          1212  1 ! IMPLICIT OUTPUTS:
;  175          1213  1 !       NONE
;  176          1214  1 !
;  177          1215  1 ! ROUTINE VALUE:
;  178          1216  1 !       Completion status.
;  179          1217  1 !
;  180          1218  1 ! SIDE EFFECTS:
;  181          1219  1 !       NONE
;  182          1220  1 !
;  183          1221  1 !--
;  184          1222  1
;  185          1223  2 BEGIN
;  186          1224  2 LOCAL
;  187          1225  2         SQH:            REF BBLOCK,     ! Pointer to SQH
;  188          1226  2         SRQ_N,                          ! Record number of SRQ record
;  189          1227  2         SRQ:            REF BBLOCK,     ! Pointer to SRQ record
;  190          1228  2         STATUS;                         ! Status return
;  191          1229  2
;  192          1230  2
;  193          1231  2 ! Allocate the queue record, and return if no more.
;  194          1232  2 !
;  195          1233  2 STATUS = ALLOCATE_RECORD( ; SRQ_N, SRQ);
;  196          1234  2 IF NOT .STATUS THEN RETURN .STATUS;
;  197          1235  2
;  198          1236  2
;  199          1237  2 ! Initialize the incomplete service record.
;  200          1238  2 !
;  201          1239  2 SRQ[SYM$B_TYPE] = SYM$K_SRQ;
;  202          1240  2 SRQ[SRQ$L_FUNCTION_CODE] = .FUNC;
;  203          1241  2 COPY_SYSID(THIS_SYSID, SRQ[SRQ$T_SENDING_SYSID]);
;  204          1242  2
;  205          1243  2
;  206          1244  2 CASE .FUNC FROM SRQ$K_START_JOB TO SRQ$K_START_SYMBIONT OF
;  207          1245  2     SET
;  208          1246  2
;  209          1247  2
;  210          1248  2     [INRANGE, OUTRANGE]:
;  211          1249  2         0;
;  212          1250  2
```

```
;  213          1251  2
;  214          1252  2      [SRQ$K_START_JOB]:
;  215          1253  3          BEGIN
;  216          1254  3          BIND
;  217          1255  3              SMQ_N       = P1,                   ! Record number of SMQ
;  218          1256  3              SMQ         = P2:   REF BBLOCK,     ! Pointer to SMQ
;  219          1257  3              SJH_N       = P3,                   ! Record number of SJH
;  220          1258  3              SJH         = P4:   REF BBLOCK;     ! Pointer to SJH
;  221          1259  3
;  222          1260  3          SRQ[SRQ$V_NO_RESPONSE] = TRUE;
;  223          1261  3          SJH[SJH$V_STARTING] = TRUE;
;  224          1262  3          COPY_SYSID(SMQ[SMQ$T_SYSID], SRQ[SRQ$T_RECEIVING_SYSID]);
;  225          1263  3          SRQ[SRQ$L_P1] = .SMQ_N;
;  226          1264  3          SRQ[SRQ$L_P2] = .SJH_N;
;  227          1265  2          END;
;  228          1266  2
;  229          1267  2
;  230          1268  2      [SRQ$K_ABORT_JOB]:
;  231          1269  3          BEGIN
;  232          1270  3          BIND
;  233          1271  3              SMQ_N       = P1,                   ! Record number of SMQ
;  234          1272  3              SMQ         = P2:   REF BBLOCK,     ! Pointer to SMQ
;  235          1273  3              SJH_N       = P3,                   ! Record number of SJH
;  236          1274  3              SJH         = P4:   REF BBLOCK;     ! Pointer to SJH
;  237          1275  3
;  238          1276  3          SJH[SJH$V_ABORTING] = TRUE;
;  239          1277  3          COPY_SYSID(SMQ[SMQ$T_SYSID], SRQ[SRQ$T_RECEIVING_SYSID]);
;  240          1278  3          SRQ[SRQ$L_P1] = .SMQ_N;
;  241          1279  3          SRQ[SRQ$L_P2] = .SJH_N;
;  242          1280  2          END;
;  243          1281  2
;  244          1282  2
;  245          1283  2      [SRQ$K_SYNCHRONIZE_JOB]:
;  246          1284  3          BEGIN
;  247          1285  3          BIND
;  248          1286  3              SJH_N       = P1,                   ! Record number of SJH
;  249          1287  3              SJH         = P2:   REF BBLOCK;     ! Pointer to SJH
;  250          1288  3
;  251          1289  3          SJH[SJH$V_SYNCHRONIZE] = TRUE;
;  252          1290  3          SRQ[SRQ$V_STALLED] = TRUE;
;  253          1291  3          COPY_SYSID(THIS_SYSID, SRQ[SRQ$T_RECEIVING_SYSID]);
;  254          1292  3          SRQ[SRQ$L_P1] = .SJH_N;
;  255          1293  2          END;
;  256          1294  2
;  257          1295  2
;  258          1296  2      [SRQ$K_START_QUEUE]:
;  259          1297  3          BEGIN
;  260          1298  3          BIND
;  261          1299  3              SMQ_N       = P1,                   ! Record number of SMQ
;  262          1300  3              SMQ         = P2:   REF BBLOCK;     ! Pointer to SMQ
;  263          1301  3
;  264          1302  3          SMQ[SMQ$V_STARTING] = TRUE;
;  265          1303  3          SMQ[SMQ$V_STOPPED] = SMQ[SMQ$V_PAUSED] = FALSE;
;  266          1304  3          COPY_SYSID(SMQ[SMQ$T_SYSID], SRQ[SRQ$T_RECEIVING_SYSID]);
;  267          1305  3          SRQ[SRQ$L_P1] = .SMQ_N;
;  268          1306  2          END;
;  269          1307  2
```

```
;  270          1308  2
;  271          1309  2      [SRQ$K_STOP_QUEUE]:
;  272          1310  3          BEGIN
;  273          1311  3          BIND
;  274          1312  3              SMQ_N       = P1,                   ! Record number of SMQ
;  275          1313  3              SMQ         = P2:   REF BBLOCK;     ! Pointer to SMQ
;  276          1314  3
;  277          1315  3          SMQ[SMQ$V_STOPPING] = TRUE;
;  278          1316  3          COPY_SYSID(SMQ[SMQ$T_SYSID], SRQ[SRQ$T_RECEIVING_SYSID]);
;  279          1317  3          SRQ[SRQ$L_P1] = .SMQ_N;
;  280          1318  2          END;
;  281          1319  2
;  282          1320  2
;  283          1321  2      [SRQ$K_PAUSE_QUEUE]:
;  284          1322  3          BEGIN
;  285          1323  3          BIND
;  286          1324  3              SMQ_N       = P1,                   ! Record number of SMQ
;  287          1325  3              SMQ         = P2:   REF BBLOCK;     ! Pointer to SMQ
;  288          1326  3
;  289          1327  3          SMQ[SMQ$V_PAUSING] = TRUE;
;  290          1328  3          COPY_SYSID(SMQ[SMQ$T_SYSID], SRQ[SRQ$T_RECEIVING_SYSID]);
;  291          1329  3          SRQ[SRQ$L_P1] = .SMQ_N;
;  292          1330  2          END;
;  293          1331  2
;  294          1332  2
;  295          1333  2      [SRQ$K_RESUME_QUEUE]:
;  296          1334  3          BEGIN
;  297          1335  3          BIND
;  298          1336  3              SMQ_N       = P1,                   ! Record number of SMQ
;  299          1337  3              SMQ         = P2:   REF BBLOCK,     ! Pointer to SMQ
;  300          1338  3              FLAGS       = P3:   BBLOCK,         ! Resume control flags
;  301          1339  3              ALIGNMENT   = P4,                   ! Alignment pages
;  302          1340  3              RELATIVE    = P5,                   ! Relative page offset
;  303          1341  3              SEARCH_LEN  = P6,                   ! Search string length
;  304          1342  3              SEARCH_ADDR = P7;                   ! Search string address
;  305          1343  3
;  306          1344  3          SMQ[SMQ$V_RESUMING] = TRUE;
;  307          1345  3          COPY_SYSID(SMQ[SMQ$T_SYSID], SRQ[SRQ$T_RECEIVING_SYSID]);
;  308          1346  3          SRQ[SRQ$L_P1] = .SMQ_N;
;  309          1347  3          SRQ[SRQ$L_P2] = .FLAGS;
;  310          1348  3          SRQ[SRQ$L_P3] = .ALIGNMENT;
;  311          1349  3          SRQ[SRQ$L_P4] = .RELATIVE;
;  312          1350  3          CH$WCHAR(.SEARCH_LEN, SRQ[SRQ$T_P5]);
;  313          1351  3          CH$MOVE(.SEARCH_LEN, .SEARCH_ADDR, SRQ[SRQ$T_P5]+1);
;  314          1352  2          END;
;  315          1353  2
;  316          1354  2
;  317          1355  2      [SRQ$K_RESET_QUEUE]:
;  318          1356  3          BEGIN
;  319          1357  3          BIND
;  320          1358  3              SMQ_N       = P1,                   ! Record number of SMQ
;  321          1359  3              SMQ         = P2:   REF BBLOCK;     ! Pointer to SMQ
;  322          1360  3
;  323          1361  3          SMQ[SMQ$V_RESETTING] = TRUE;
;  324          1362  3          COPY_SYSID(SMQ[SMQ$T_SYSID], SRQ[SRQ$T_RECEIVING_SYSID]);
;  325          1363  3          SRQ[SRQ$L_P1] = .SMQ_N;
;  326          1364  2          END;
```

```
;  327          1365  2
;  328          1366  2
;  329          1367  2      [SRQ$K_BROADCAST_MESSAGE]:
;  330          1368  3          BEGIN
;  331          1369  3          BIND
;  332          1370  3              SYSID       = P1,
;  333          1371  3              USERNAME    = P2:   REF VECTOR[,BYTE],
;  334          1372  3              LENGTH      = P3,
;  335          1373  3              ADDRESS     = P4;
;  336          1374  3
;  337          1375  3          SRQ[SRQ$V_NO_RESPONSE] = TRUE;
;  338          1376  3          COPY_SYSID(.SYSID, SRQ[SRQ$T_RECEIVING_SYSID]);
;  339          1377  3          CH$MOVE(SRQ$S_BRDCST_USERNAME, .USERNAME, SRQ[SRQ$T_BRDCST_USERNAME]);
;  340          1378  3          SRQ[SRQ$W_BRDCST_LENGTH] = .LENGTH;
;  341          1379  3          CH$MOVE(.LENGTH, .ADDRESS, SRQ[SRQ$T_BRDCST_TEXT]);
;  342          1380  2          END;
;  343          1381  2
;  344          1382  2
;  345          1383  2      [SRQ$K_DELETE_FILES]:
;  346          1384  3          BEGIN
;  347          1385  3          BIND
;  348          1386  3              SJH         = P1:   REF BBLOCK,       Pointer to SJH
;  349          1387  3              SQR_N       = P2;                    ! Record number of SQR
;  350          1388  3
;  351          1389  3          SRQ[SRQ$V_NO_RESPONSE] = TRUE;
;  352          1390  3          COPY_SYSID(SJH[SJH$T_SYSID], SRQ[SRQ$T_RECEIVING_SYSID]);
;  353          1391  3          SRQ[SRQ$L_P1] = .SQR_N;
;  354          1392  2          END;
;  355          1393  2
;  356          1394  2
;  357          1395  2      [SRQ$K_START_SYMBIONT]:
;  358          1396  3          BEGIN
;  359          1397  3          BIND
;  360          1398  3              SMQ_N       = P1,                    ! Record number of SMQ
;  361          1399  3              SMQ         = P2:   REF BBLOCK;      ! Pointer to SMQ
;  362          1400  3
;  363          1401  3          SRQ[SRQ$V_STALLED] = TRUE;
;  364          1402  3          COPY_SYSID(SMQ[SMQ$T_SYSID], SRQ[SRQ$T_RECEIVING_SYSID]);
;  365          1403  3          SRQ[SRQ$L_P1] = .SMQ_N;
;  366          1404  2          END;
;  367          1405  2
;  368          1406  2
;  369          1407  2      TES;
;  370          1408  2
;  371          1409  2
;  372          1410  2  IF NOT .SRQ[SRQ$V_NO_RESPONSE]
;  373          1411  2  THEN
;  374          1412  2      CREATE_SRB(SRQ[SRQ$T_SRB]);
;  375          1413  2
;  376          1414  2
;  377          1415  2  ! If services of another job controller are required, signal it.
;  378          1416  2  !
;  379          1417  3  IF SYSID_NEQ(THIS_SYSID, SRQ[SRQ$T_RECEIVING_SYSID])
;  380          1418  2  AND NOT .SRQ[SRQ$V_STALLED]
;  381          1419  2  THEN
;  382          1420  2      ENTER_REMOTE_REQUEST(SRQ[SRQ$T_RECEIVING_SYSID]);
;  383          1421  2
```

```
;  384          1422  2
;  385          1423  2 ! Enqueue the record to the incomplete service list.
;  386          1424  2 !
;  387          1425  2 SQH = READ_RECORD(SQH$K_RECNO);
;  388          1426  2 SRQ[SYM$L_LINK] = .SQH[SQH$L_INCOMPLETE_SERVICE_LIST];
;  389          1427  2 SQH[SQH$L_INCOMPLETE_SERVICE_LIST] = .SRQ_N;
;  390          1428  2 REWRITE_RECORD(.SRQ_N);
;  391          1429  2 REWRITE_RECORD(SQH$K_RECNO);
;  392          1430  2
;  393          1431  2
;  394          1432  2 ! Return 0 to indicate that the service is incomplete.
;  395          1433  2 !
;  396          1434  2 0
;  397          1435  1 END;

                                .TITLE  ASYNCHRON Asynchronous service management
                                .IDENT  \V04-002\

                                .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   8
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
00F82 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   8
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   8
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  ABORT_EXECUTION
                        .EXTRN  AFTER_AST, ALLOCATE_MEMORY
                        .EXTRN  ALLOCATE_RECORD
                        .EXTRN  BROADCAST_MESSAGE
                        .EXTRN  COMPLETE_JOB, CREATE_SRB
                        .EXTRN  DEALLOCATE_MEMORY
                        .EXTRN  DEALLOCATE_RECORD
                        .EXTRN  DELETE_FILES, FIND_PENDING_JOBS
                        .EXTRN  FLUSH_RECORD, LOCK_QUEUE_FILE
                        .EXTRN  PAUSE_EXECUTION
                        .EXTRN  READ_RECORD, RELEASE_RECORD
                        .EXTRN  RESET_EXECUTOR_QUEUE
                        .EXTRN  RESUME_EXECUTION
                        .EXTRN  REWRITE_RECORD, SCHEDULE_NONAST
                        .EXTRN  SEND_SERVICE_RESPONSE_MESSAGE
                        .EXTRN  START_EXECUTION
                        .EXTRN  START_SYMBIONT_STREAM
                        .EXTRN  STOP_SYMBIONT_STREAM
                        .EXTRN  UNLOCK_QUEUE_FILE
                        .EXTRN  UPDATE_GETQUI_DATA

                        .PSECT  CODE,NOWRT,2

0CFC 00000              .ENTRY  CREATE_SRQ_RECORD, Save R2,R3,R4,R5,R6,R7,- ; 1194
                                R10,R11                                     ;
```

```
                                    57 00000000G   EF   9E 00002          MOVAB   REWRITE_RECORD, R7                                ;
                                    56 00000000'   EF   9E 00009          MOVAB   THIS_SYSID, R6                                   ;
                          00000000G EF             00   FB 00010          CALLS   #0, ALLOCATE_RECORD                      ;  1233
                                    01             50   E8 00017          BLBS    STATUS, 1$                               ;  1234
                                                   04   0001A            RET                                                      ;
                              04    AB             09   90 0001B  1$:     MOVB    #9, 4(SRQ)                               ;  1239
                              0C    AB       04    AC   D0 0001F          MOVL    FUNC, 12(SRQ)                            ;  1240
                              14    AB             66   D0 00024          MOVL    THIS_SYSID, 20(SRQ)                      ;  1241
                              18    AB       04    A6   B0 00028          MOVW    THIS_SYSID+4, 24(SRQ)                            ;
                        0B          01       04    AC   CF 0002D          CASEL   FUNC, #1, #11                            ;  1244
          0064          004C          0028          001A     00032  2$:     .WORD   3$-2$,-                                  ;
          00BA          0088          007E          0074     0003A          .WORD   4$-2$,-                                  ;
          0104          00E9          011D          00C4     00042          .WORD   7$-2$,-                                  ;
                                                                              .WORD   8$-2$,-                                  ;
                                                                              .WORD   9$-2$,-                                  ;
                                                                              .WORD   10$-2$,-                                 ;
                                                                              .WORD   11$-2$,-                                 ;
                                                                              .WORD   13$-2$,-                                 ;
                                                                              .WORD   15$-2$,-                                 ;
                                                                              .WORD   20$-2$,-                                 ;
                                                                              .WORD   16$-2$,-                                 ;
                                                                              .WORD   17$-2$                                   ;
                                                 30   11 0004A          BRB     6$                                       ;
                              10    AB             01   88 0004C  3$:     BISB2   #1, 16(SRQ)                              ;  1260
                                    50       14    AC   D0 00050          MOVL    SJH, R0                                  ;  1261
                              11    A0             10   88 00054          BISB2   #16, 17(R0)                                      ;
                                                   08   11 00058          BRB     5$                                       ;  1262
                                    50       14    AC   D0 0005A  4$:     MOVL    SJH, R0                                  ;  1276
                              10    A0             02   88 0005E          BISB2   #2, 16(R0)                                       ;
                                    50       0C    AC   D0 00062  5$:     MOVL    SMQ, R0                                  ;  1277
                              1A    AB       0106  C0   D0 00066          MOVL    262(R0), 26(SRQ)                                 ;
                              1E    AB       010A  C0   B0 0006C          MOVW    266(R0), 30(SRQ)                                 ;
                              20    AB         08  AC   D0 00072          MOVL    SMQ_N, 32(SRQ)                           ;  1278
                              24    AB         10  AC   D0 00077          MOVL    SJH_N, 36(SRQ)                           ;  1279
                                                   6C   11 0007C  6$:     BRB     12$                                      ;  1244
                                    50       0C    AC   D0 0007E  7$:     MOVL    SJH, R0                                  ;  1289
                              11    A0             20   88 00082          BISB2   #32, 17(R0)                                      ;
                              10    AB             02   88 00086          BISB2   #2, 16(SRQ)                              ;  1290
                              1A    AB             66   D0 0008A          MOVL    THIS_SYSID, 26(SRQ)                      ;  1291
                              1E    AB       04    A6   B0 0008E          MOVW    THIS_SYSID+4, 30(SRQ)                            ;
                                          00B4     31   00093            BRW     19$                                      ;  1292
                                    50       0C    AC   D0 00096  8$:     MOVL    SMQ, R0                                  ;  1302
                              11    A0             01   88 0009A          BISB2   #1, 17(R0)                                       ;
                              10    A0       0204  8F   AA 0009E          BICW2   #516, 16(R0)                             ;  1303
                                                   4E   11 000A4          BRB     14$                                      ;  1304
                                    50       0C    AC   D0 000A6  9$:     MOVL    SMQ, R0                                  ;  1315
                              11    A0             04   88 000AA          BISB2   #4, 17(R0)                                       ;
                                                   44   11 000AE          BRB     14$                                      ;  1316
                                    50       0C    AC   D0 000B0  10$:    MOVL    SMQ, R0                                  ;  1327
                              10    A0             08   88 000B4          BISB2   #8, 16(R0)                                       ;
                                                   3A   11 000B8          BRB     14$                                      ;  1328
                                    50       0C    AC   D0 000BA  11$:    MOVL    SMQ, R0                                  ;  1344
                              10    A0             40   8F   88 000BE          BISB2   #64, 16(R0)                              ;
                              1A    AB       0106  C0   D0 000C3          MOVL    262(R0), 26(SRQ)                         ;  1345
                              1E    AB       010A  C0   B0 000C9          MOVW    266(R0), 30(SRQ)                                 ;
                              20    AB         08  AC   D0 000CF          MOVL    SMQ_N, 32(SRQ)                           ;  1346
                              24    AB         10  AC   7D 000D4          MOVQ    FLAGS, 36(SRQ)                           ;  1347
```

```
                        2C  AB      18  AC  D0 000D9          MOVL    RELATIVE, 44(SRQ)                                   ; 1349
                        30  AB      1C  AC  90 000DE          MOVB    SEARCH_LEN, 48(SRQ)                                 ; 1350
                31  AB  20  BC      1C  AC  28 000E3          MOVC3   SEARCH_LEN, @SEARCH_ADDR, 49(SRQ)                   ; 1351
                                        63  11 000EA 12$:     BRB     20$                                                 ; 1244
                            50      0C  AC  D0 000EC 13$:     MOVL    SMQ, R0                                             ; 1361
                        10  A0          20  88 000F0         BISB2   #32, 16(R0)
                                        48  11 000F4 14$:     BRB     18$                                                 ; 1362
                        10  AB          01  88 000F6 15$:     BISB2   #1, 16(SRQ)                                         ; 1375
                            50      08  AC  D0 000FA         MOVL    SYSID, R0                                           ; 1376
                        1A  AB          60  D0 000FE         MOVL    (R0), 26(SRQ)
                        1E  AB      04  A0  B0 00102         MOVW    4(R0), 30(SRQ)
                20  AB  0C  BC          0C  28 00107         MOVC3   #12, @USERNAME, 32(SRQ)                             ; 1377
                        40  AB      10  AC  B0 0010D         MOVW    LENGTH, 64(SRQ)                                     ; 1378
                42  AB  14  BC      10  AC  28 00112         MOVC3   LENGTH, @ADDRESS, 66(SRQ)                           ; 1379
                                        34  11 00119         BRB     20$                                                 ; 1244
                        10  AB          01  88 0011B 16$:     BISB2   #1, 16(SRQ)                                         ; 1389
                            50      08  AC  D0 0011F         MOVL    SJH, R0                                             ; 1390
                        1A  AB  0160C   C0  D0 00123         MOVL    364(R0), 26(SRQ)
                        1E  AB  01710   C0  B0 00129         MOVW    368(R0), 30(SRQ)
                        20  AB      0C  AC  D0 0012F         MOVL    SQR_N, 32(SRQ)                                      ; 1391
                                        19  11 00134         BRB     20$                                                 ; 1244
                        10  AB          02  88 00136 17$:     BISB2   #2, 16(SRQ)                                         ; 1401
                            50      0C  AC  D0 0013A         MOVL    SMQ, R0                                             ; 1402
                        1A  AB  01060   C0  D0 0013E 18$:     MOVL    262(R0), 26(SRQ)
                        1E  AB  010A0   C0  B0 00144         MOVW    266(R0), 30(SRQ)
                        20  AB      08  AC  D0 0014A 19$:     MOVL    SMQ_N, 32(SRQ)                                      ; 1403
                            0A      10  AB  E8 0014F 20$:     BLBS    16(SRQ), 21$                                        ; 1410
                                    70  AB  9F 00153         PUSHAB  112(SRQ)                                            ; 1412
                00000000G   EF          01  FB 00156         CALLS   #1, CREATE_SRB
                        1A  AB          66  D1 0015D 21$:     CMPL    THIS_SYSID, 26(SRQ)                                 ; 1417
                                        07  12 00161         BNEQ    22$
                        1E  AB      04  A6  B1 00163         CMPW    THIS_SYSID+4, 30(SRQ)
                                        0D  13 00168         BEQL    23$
            08          10  AB          01  E0 0016A 22$:     BBS     #1, 16(SRQ), 23$                                    ; 1418
                                    1A  AB  9F 0016F         PUSHAB  26(SRQ)                                             ; 1420
                    0000V   CF          01  FB 00172         CALLS   #1, ENTER_REMOTE_REQUEST
                                        01  DD 00177 23$:     PUSHL   #1                                                  ; 1425
                00000000G   EF          01  FB 00179         CALLS   #1, READ_RECORD
                            6B      44  A0  D0 00180         MOVL    68(SQH), (SRQ)                                      ; 1426
                        44  A0          5A  D0 00184         MOVL    SRQ_N, 68(SQH)                                      ; 1427
                                        5A  DD 00188         PUSHL   SRQ_N                                               ; 1428
                                    67      01  FB 0018A         CALLS   #1, REWRITE_RECORD
                                        01  DD 0018D         PUSHL   #1                                                  ; 1429
                                    67      01  FB 0018F         CALLS   #1, REWRITE_RECORD
                                        50  D4 00192         CLRL    R0                                                  ; 1435
                                            04 00194         RET                                                         ;

; Routine Size:  405 bytes,    Routine Base:  CODE + 0000
```

```
;  399     1436  1 ROUTINE PROCESS_REMOTE_SERVICES(SRQ;NEXT_ACTION): L_OUTPUT_1=
;  400     1437  1
;  401     1438  1 !++
;  402     1439  1 !
;  403     1440  1 ! FUNCTIONAL DESCRIPTION:
;  404     1441  1 !       This routine processes a remote service directed to this node.
;  405     1442  1 !
;  406     1443  1 ! INPUT PARAMETERS:
;  407     1444  1 !       SRQ             - Pointer to SRQ.
;  408     1445  1 !
;  409     1446  1 ! IMPLICIT INPUTS:
;  410     1447  1 !       NONE
;  411     1448  1 !
;  412     1449  1 ! OUTPUT PARAMETERS:
;  413     1450  1 !       NEXT_ACTION     - Code identifying the next action.
;  414     1451  1 !
;  415     1452  1 ! IMPLICIT OUTPUTS:
;  416     1453  1 !       NONE
;  417     1454  1 !
;  418     1455  1 ! ROUTINE VALUE:
;  419     1456  1 !       Completion status.
;  420     1457  1 !
;  421     1458  1 ! SIDE EFFECTS:
;  422     1459  1 !       NONE
;  423     1460  1 !
;  424     1461  1 !--
;  425     1462  1
;  426     1463  2 BEGIN
;  427     1464  2 MAP
;  428     1465  2         SRQ:            REF BBLOCK;     ! Pointer to SRQ
;  429     1466  2 LOCAL
;  430     1467  2         STATUS;                         ! Status of the request
;  431     1468  2
;  432     1469  2
;  433     1470  2 STATUS = SS$_NORMAL;
;  434     1471  2
;  435     1472  2
;  436     1473  2 CASE .SRQ[SRQ$L_FUNCTION_CODE] FROM SRQ$K_START_JOB TO SRQ$K_DELETE_FILES OF
;  437     1474  2     SET
;  438     1475  2
;  439     1476  2
;  440     1477  2     [INRANGE, OUTRANGE]:
;  441     1478  2         NEXT_ACTION = K_COMPLETE;
;  442     1479  2
;  443     1480  2
;  444     1481  2     [SRQ$K_START_JOB]:
;  445     1482  3         BEGIN
;  446     1483  3         LOCAL
;  447     1484  3             SMQ_N,                      ! Record number of SMQ
;  448     1485  3             SMQ:        REF BBLOCK,     ! Pointer to SMQ
;  449     1486  3             SJH_NP,                     ! Record number of predecessor of SJH
;  450     1487  3             SJH_P:      REF BBLOCK,     ! Predecessor of SJH
;  451     1488  3             SJH_N,                      ! Record number of SJH
;  452     1489  3             SJH:        REF BBLOCK;     ! Pointer to SJH
;  453     1490  3
;  454     1491  3
;  455     1492  3         SMQ = READ_RECORD(SMQ_N = SJH_NP = .SRQ[SRQ$L_P1]);
```

```
;  456          1493  3          SJH_N = .SMQ[SMQ$L_CURRENT_LIST];
;  457          1494  3          WHILE .SJH_N NEQ 0 DO
;  458          1495  4              BEGIN
;  459          1496  4              SJH = READ_RECORD(.SJH_N);
;  460          1497  4              IF .SJH_N EQL .SRQ[SRQ$L_P2]
;  461          1498  4              THEN
;  462          1499  5                  BEGIN
;  463          1500  5                  SJH[SJH$V_STARTING] = FALSE;
;  464          1501  5                  STATUS = START_EXECUTION(
;  465          1502  5                      .SMQ_N, .SMQ,
;  466          1503  5                      .SJH_N, .SJH);
;  467          1504  5                  IF NOT .STATUS
;  468          1505  5                  THEN
;  469          1506  6                      BEGIN
;  470          1507  6                      UPDATE_GETQUI_DATA(.SJH_N, .SJH);
;  471          1508  6                      SMQ[SMQ$B_CURRENT_JOB_COUNT] = .SMQ[SMQ$B_CURRENT_JOB_COUNT] - 1;
;  472          1509  6                      IF .SJH_NP EQL .SMQ_N
;  473          1510  6                      THEN
;  474          1511  7                          BEGIN
;  475          1512  7                          SMQ[SMQ$L_CURRENT_LIST] = .SJH[SYM$L_LINK];
;  476          1513  7                          IF .SJH[SYM$L_LINK] EQL 0 THEN SMQ[SMQ$L_CURRENT_LIST_END] = 0;
;  477          1514  7                          END
;  478          1515  6                      ELSE
;  479          1516  7                          BEGIN
;  480          1517  7                          SJH_P[SYM$L_LINK] = .SJH[SYM$L_LINK];
;  481          1518  7                          IF .SJH[SYM$L_LINK] EQL 0 THEN SMQ[SMQ$L_CURRENT_LIST_END] = .SJH_NP;
;  482          1519  7                          REWRITE_RECORD(.SJH_NP);
;  483          1520  6                          END;
;  484          1521  6                      SJH[SJH$L_CONDITION_1] = .STATUS;
;  485          1522  6                      COMPLETE_JOB(.SJH_N, .SJH, .SMQ, 0);
;  486          1523  6                      FIND_PENDING_JOBS(.SMQ_N, .SMQ);
;  487          1524  6                      END
;  488          1525  5                  ELSE
;  489          1526  5                      REWRITE_RECORD(.SJH_N);
;  490          1527  5
;  491          1528  5                  REWRITE_RECORD(.SMQ_N);
;  492          1529  5                  EXITLOOP;
;  493          1530  4                  END;
;  494          1531  4              IF .SJH_NP NEQ .SMQ_N THEN RELEASE_RECORD(.SJH_NP);
;  495          1532  4              SJH_NP = .SJH_N;
;  496          1533  4              SJH_P = .SJH;
;  497          1534  4              SJH_N = .SJH[SYM$L_LINK];
;  498          1535  3              END;
;  499          1536  3          NEXT_ACTION = K_DEALLOCATE;
;  500          1537  2          END;
;  501          1538  2
;  502          1539  2
;  503          1540  2      [SRQ$K_ABORT_JOB]:
;  504          1541  3          BEGIN
;  505          1542  3          LOCAL
;  506          1543  3              SMQ_N,                      ! Record number of SMQ
;  507          1544  3              SMQ:        REF BBLOCK,     ! Pointer to SMQ
;  508          1545  3              SJH_N,                      ! Record number of SJH
;  509          1546  3              SJH_NS,                     ! Successor of SJH
;  510          1547  3              SJH:        REF BBLOCK;     ! Pointer to SJH
;  511          1548  3
;  512          1549  3
```

```
;  513     1550  3          SMQ = READ_RECORD(SMQ_N = .SRQ[SRQ$L_P1]);
;  514     1551  3          SJH_N = .SMQ[SMQ$L_CURRENT_LIST];
;  515     1552  3          WHILE .SJH_N NEQ 0 DO
;  516     1553  4              BEGIN
;  517     1554  4              SJH = READ_RECORD(.SJH_N);
;  518     1555  4              IF .SJH_N EQL .SRQ[SRQ$L_P2]
;  519     1556  4              THEN
;  520     1557  5                  BEGIN
;  521     1558  5                  SJH[SJH$V_ABORTING] = FALSE;
;  522     1559  5                  STATUS = ABORT_EXECUTION(
;  523     1560  5                      .SMQ_N, .SMQ,
;  524     1561  5                      .SJH_N, .SJH);
;  525     1562  5                  REWRITE_RECORD(.SJH_N);
;  526     1563  5                  EXITLOOP;
;  527     1564  4                  END;
;  528     1565  4              SJH_NS = .SJH[SYM$L_LINK];
;  529     1566  4              RELEASE_RECORD(.SJH_N);
;  530     1567  4              SJH_N = .SJH_NS;
;  531     1568  3              END;
;  532     1569  3          NEXT_ACTION = K_COMPLETE;
;  533     1570  2          END;
;  534     1571  2
;  535     1572  2
;  536     1573  2      [SRQ$K_START_QUEUE]:
;  537     1574  3          BEGIN
;  538     1575  3          LOCAL
;  539     1576  3              SMQ_N,                          ! Record number of SMQ
;  540     1577  3              SMQ:                REF BBLOCK; ! Pointer to SMQ
;  541     1578  3
;  542     1579  3
;  543     1580  3          SMQ = READ_RECORD(SMQ_N = .SRQ[SRQ$L_P1]);
;  544     1581  3          STATUS = START_SYMBIONT_STREAM(.SMQ_N, .SMQ);
;  545     1582  3          IF NOT .STATUS
;  546     1583  3          THEN
;  547     1584  4              BEGIN
;  548     1585  4              SMQ[SMQ$V_STARTING] = FALSE;
;  549     1586  4              SMQ[SMQ$V_STOPPED] = TRUE;
;  550     1587  4              NEXT_ACTION = K_COMPLETE;
;  551     1588  4              END
;  552     1589  3          ELSE
;  553     1590  4              BEGIN
;  554     1591  4              SRQ[SRQ$L_FUNCTION_CODE] = SRQ$K_START_SYMBIONT;
;  555     1592  4              SRQ[SRQ$V_STALLED] = TRUE;
;  556     1593  4              NEXT_ACTION = K_REWRITE;
;  557     1594  3              END;
;  558     1595  3          REWRITE_RECORD(.SMQ_N);
;  559     1596  2          END;
;  560     1597  2
;  561     1598  2
;  562     1599  2      [SRQ$K_STOP_QUEUE]:
;  563     1600  3          BEGIN
;  564     1601  3          LOCAL
;  565     1602  3              SMQ_N,                          ! Record number of SMQ
;  566     1603  3              SMQ:                REF BBLOCK; ! Pointer to SMQ
;  567     1604  3
;  568     1605  3
;  569     1606  3          SMQ = READ_RECORD(SMQ_N = .SRQ[SRQ$L_P1]);
```

```
;  570          1607  3          SMQ[SMQ$V_STOPPING] = FALSE;
;  571          1608  3          STOP_SYMBIONT_STREAM(.SMQ_N, .SMQ);
;  572          1609  3          REWRITE_RECORD(.SMQ_N);
;  573          1610  3          NEXT_ACTION = K_COMPLETE;
;  574          1611  2          END;
;  575          1612  2
;  576          1613  2
;  577          1614  2      [SRQ$K_PAUSE_QUEUE]:
;  578          1615  3          BEGIN
;  579          1616  3          LOCAL
;  580          1617  3              SMQ_N,                              ! Record number of SMQ
;  581          1618  3              SMQ:                REF BBLOCK;     ! Pointer to SMQ
;  582          1619  3
;  583          1620  3
;  584          1621  3          SMQ = READ_RECORD(SMQ_N = .SRQ[SRQ$L_P1]);
;  585          1622  3          SMQ[SMQ$V_PAUSING] = FALSE;
;  586          1623  3          STATUS = PAUSE_EXECUTION(.SMQ_N, .SMQ);
;  587          1624  3          IF NOT .STATUS THEN FIND_PENDING_JOBS(.SMQ_N, .SMQ);
;  588          1625  3          REWRITE_RECORD(.SMQ_N);
;  589          1626  3          NEXT_ACTION = K_COMPLETE;
;  590          1627  2          END;
;  591          1628  2
;  592          1629  2
;  593          1630  2      [SRQ$K_RESUME_QUEUE]:
;  594          1631  3          BEGIN
;  595          1632  3          LOCAL
;  596          1633  3              SMQ_N,                              ! Record number of SMQ
;  597          1634  3              SMQ:                REF BBLOCK;     ! Pointer to SMQ
;  598          1635  3
;  599          1636  3
;  600          1637  3          SMQ = READ_RECORD(SMQ_N = .SRQ[SRQ$L_P1]);
;  601          1638  3          SMQ[SMQ$V_RESUMING] = FALSE;
;  602          1639  3          STATUS = RESUME_EXECUTION(
;  603          1640  3              .SMQ_N, .SMQ,
;  604          1641  3              .SRQ[SRQ$L_P2], .SRQ[SRQ$L_P3], .SRQ[SRQ$L_P4],
;  605          1642  3              CH$RCHAR(SRQ[SRQ$T_P5]), SRQ[SRQ$T_P5]+1);
;  606          1643  3          FIND_PENDING_JOBS(.SMQ_N, .SMQ);
;  607          1644  3          REWRITE_RECORD(.SMQ_N);
;  608          1645  3          NEXT_ACTION = K_COMPLETE;
;  609          1646  2          END;
;  610          1647  2
;  611          1648  2
;  612          1649  2      [SRQ$K_RESET_QUEUE]:
;  613          1650  3          BEGIN
;  614          1651  3          LOCAL
;  615          1652  3              SMQ_N,                              ! Record number of SMQ
;  616          1653  3              SMQ:                REF BBLOCK;     ! Pointer to SMQ
;  617          1654  3
;  618          1655  3
;  619          1656  3          SMQ[SMQ$V_RESETTING] = FALSE;
;  620          1657  3          SMQ = READ_RECORD(SMQ_N = .SRQ[SRQ$L_P1]);
;  621          1658  3          RESET_EXECUTOR_QUEUE(.SMQ_N, .SMQ);
;  622          1659  3          REWRITE_RECORD(.SMQ_N);
;  623          1660  3          NEXT_ACTION = K_COMPLETE;
;  624          1661  2          END;
;  625          1662  2
;  626          1663  2
```

```
;  627          1664  2      [SRQ$K_BROADCAST_MESSAGE]:
;  628          1665  3          BEGIN
;  629          1666  3          BROADCAST_MESSAGE(
;  630          1667  3              THIS_SYSID,
;  631          1668  3              SRQ[SRQ$T_BRDCST_USERNAME],
;  632          1669  3              .SRQ[SRQ$W_BRDCST_LENGTH],
;  633          1670  3              SRQ[SRQ$T_BRDCST_TEXT]);
;  634          1671  3          NEXT_ACTION = K_DEALLOCATE;
;  635          1672  2          END;
;  636          1673  2
;  637          1674  2
;  638          1675  2      [SRQ$K_RESPONSE]:
;  639          1676  3          BEGIN
;  640          1677  3          SEND_SERVICE_RESPONSE_MESSAGE(
;  641          1678  3              SRQ[SRQ$T_SRB],
;  642          1679  3              .SRQ[SRQ$L_P1]);
;  643          1680  3          NEXT_ACTION = K_DEALLOCATE;
;  644          1681  2          END;
;  645          1682  2
;  646          1683  2
;  647          1684  2      [SRQ$K_DELETE_FILES]:
;  648          1685  3          BEGIN
;  649          1686  3          DELETE_FILES(.SRQ[SRQ$L_P1]);
;  650          1687  3          NEXT_ACTION = K_DEALLOCATE;
;  651          1688  2          END;
;  652          1689  2
;  653          1690  2
;  654          1691  2      TES;
;  655          1692  2
;  656          1693  2
;  657          1694  2 .STATUS
;  658          1695  1 END;
; INFO#250                 L1:1517
; Referenced LOCAL symbol SJH_P is probably not initialized
; INFO#250                 L1:1656
; Referenced LOCAL symbol SMQ is probably not initialized

                                                07FC 00000 PROCESS_REMOTE_SERVICES:
                                                           .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10          ; 1436
                                    5A 00000000G   EF  9E 00002    MOVAB   READ_RECORD, R10                  ;
                                    58             01  D0 00009    MOVL    #1, STATUS                        ; 1470
                                    52         04  AC  D0 0000C    MOVL    SRQ, R2                           ; 1473
                      0A            01         0C  A2  CF 00010    CASEL   12(R2), #1, #10                   ;
          0126      01FF          00D7           0019    00015 1$:  .WORD   2$-1$,-                           ;
          01DE      01A3          0180           0166    0001D      12$-1$,-                                  ;
                    022B          021C           0203    00025      28$-1$,-                                  ;
                                                                    17$-1$,-                                  ;
                                                                    20$-1$,-                                  ;
                                                                    21$-1$,-                                  ;
                                                                    22$-1$,-                                  ;
                                                                    25$-1$,-                                  ;
                                                                    29$-1$,-                                  ;
                                                                    30$-1$,-                                  ;
```

```
                                                        31$-1$
                        01E6  31 0002B          BRW     28$                             ; 1478
                    56    20  A2 D0 0002E 2$:   MOVL    32(R2), SJH_NP                  ; 1492
                    57        56 D0 00032       MOVL    SJH_NP, SMQ_N
                              56 DD 00035       PUSHL   SJH_NP
                    6A        01 FB 00037       CALLS   #1, READ_RECORD
                    53        50 D0 0003A       MOVL    R0, SMQ
                    55    48  A3 D0 0003D       MOVL    72(SMQ), SJH_N                  ; 1493
                              03 12 00041 3$:   BNEQ    4$                              ; 1494
                          0204  31 00043        BRW     32$
                              55 DD 00046 4$:   PUSHL   SJH_N                           ; 1496
                    6A        01 FB 00048       CALLS   #1, READ_RECORD
                    54        50 D0 0004B       MOVL    R0, SJH
                24  A2        55 D1 0004E       CMPL    SJH_N, 36(R2)                   ; 1497
                              7E 12 00052       BNEQ    10$
                11  A4        10 8A 00054       BICB2   #16, 17(SJH)                    ; 1500
                              54 DD 00058       PUSHL   SJH                             ; 1503
                              28 BB 0005A       PUSHR   #^M<R3,R5>                      ; 1502
                              57 DD 0005C       PUSHL   SMQ_N
00000000G       EF            04 FB 0005E       CALLS   #4, START_EXECUTION
                    58        50 D0 00065       MOVL    R0, STATUS
                    52        58 E8 00068       BLBS    STATUS, 8$                      ; 1504
                              54 DD 0006B       PUSHL   SJH                             ; 1507
                              55 DD 0006D       PUSHL   SJH_N
00000000G       EF            02 FB 0006F       CALLS   #2, UPDATE_GETQUI_DATA
                        0115  C3 97 00076       DECB    277(SMQ)                        ; 1508
                    57        56 D1 0007A       CMPL    SJH_NP, SMQ_N                   ; 1509
                              0B 12 0007D       BNEQ    5$
                48  A3        64 D0 0007F       MOVL    (SJH), 72(SMQ)                  ; 1512
                              17 12 00083       BNEQ    7$                              ; 1513
                          4C  A3 D4 00085       CLRL    76(SMQ)
                              12 11 00088       BRB     7$                              ; 1509
                    69        64 D0 0008A 5$:   MOVL    (SJH), (SJH_P)                  ; 1517
                              04 12 0008D       BNEQ    6$                              ; 1518
                          4C  A3 56 D0 0008F    MOVL    SJH_NP, 76(SMQ)
                              56 DD 00093 6$:   PUSHL   SJH_NP                          ; 1519
00000000G       EF            01 FB 00095       CALLS   #1, REWRITE_RECORD
            00DC  C4          58 D0 0009C 7$:   MOVL    STATUS, 220(SJH)                ; 1521
                              7E D4 000A1       CLRL    -(SP)                           ; 1522
                              53 DD 000A3       PUSHL   SMQ
                              54 DD 000A5       PUSHL   SJH
                              55 DD 000A7       PUSHL   SJH_N
00000000G       EF            04 FB 000A9       CALLS   #4, COMPLETE_JOB
                              53 DD 000B0       PUSHL   SMQ                             ; 1523
                              57 DD 000B2       PUSHL   SMQ_N
00000000G       EF            02 FB 000B4       CALLS   #2, FIND_PENDING_JOBS
                              09 11 000BB       BRB     9$                              ; 1504
                              55 DD 000BD 8$:   PUSHL   SJH_N                           ; 1526
00000000G       EF            01 FB 000BF       CALLS   #1, REWRITE_RECORD
                              57 DD 000C6 9$:   PUSHL   SMQ_N                           ; 1528
00000000G       EF            01 FB 000C8       CALLS   #1, REWRITE_RECORD
                          0178  31 000CF        BRW     32$                             ; 1499
                    57        56 D1 000D2 10$:  CMPL    SJH_NP, SMQ_N                   ; 1531
                              09 13 000D5       BEQL    11$
                              56 DD 000D7       PUSHL   SJH_NP
00000000G       EF            01 FB 000D9       CALLS   #1, RELEASE_RECORD
                    56        55 D0 000E0 11$:  MOVL    SJH_N, SJH_NP                   ; 1532
```

```
                        59              54  D0 000E3          MOVL    SJH, SJH_P                      ; 1533
                        55              64  D0 000E6          MOVL    (SJH), SJH_N                    ; 1534
                                     FF55  31 000E9          BRW     3$                              ; 1494
                        57        20    A2  D0 000EC 12$:     MOVL    32(R2), SMQ_N                   ; 1550
                                        57  DD 000F0          PUSHL   SMQ_N                           ;
                        6A              01  FB 000F2          CALLS   #1, READ_RECORD                 ;
                        55              50  D0 000F5          MOVL    R0, SMQ                         ;
                        54        4R    A5  D0 000F8          MOVL    72(SMQ), SJH_N                  ; 1551
                                        03  12 000FC 13$:     BNEQ    14$                             ; 1552
                                      0113  31 000FE          BRW     28$                             ;
                                        54  DD 00101 14$:     PUSHL   SJH_N                           ; 1554
                        6A              01  FB 00103          CALLS   #1, READ_RECORD                 ;
                        53              50  D0 00106          MOVL    R0, SJH                         ;
                     24 A2              54  D1 00109          CMPL    SJH_N, 36(R2)                   ; 1555
                                        1B  12 0010D          BNEQ    16$                             ;
                     10 A3              02  8A 0010F          BICB2   #2, 16(SJH)                     ; 1558
                                        53  DD 00113          PUSHL   SJH                             ; 1561
                                        54  DD 00115          PUSHL   SJH_N                           ;
                                        55  DD 00117          PUSHL   SMQ                             ; 1560
                                        57  DD 00119          PUSHL   SMQ_N                           ;
                00000000G               EF  04 FB 0011B       CALLS   #4, ABORT_EXECUTION             ;
                        58              50  D0 00122          MOVL    R0, STATUS                      ;
                                        54  DD 00125 15$:     PUSHL   SJH_N                           ; 1562
                                      00E3  31 00127          BRW     27$                             ;
                        56              63  D0 0012A 16$:     MOVL    (SJH), SJH_NS                   ; 1565
                                        54  DD 0012D          PUSHL   SJH_N                           ; 1566
                00000000G               EF  01 FB 0012F       CALLS   #1, RELEASE_RECORD              ;
                        54              56  D0 00136          MOVL    SJH_NS, SJH_N                   ; 1567
                                        C1  11 00139          BRB     13$                             ; 1552
                        54        20    A2  D0 0013B 17$:     MOVL    32(R2), SMQ_N                   ; 1580
                                        54  DD 0013F          PUSHL   SMQ_N                           ;
                        6A              01  FB 00141          CALLS   #1, READ_RECORD                 ;
                        53              50  D0 00144          MOVL    R0, SMQ                         ;
                                        53  DD 00147          PUSHL   SMQ                             ; 1581
                                        54  DD 00149          PUSHL   SMQ_N                           ;
                00000000G               EF  02 FB 0014B       CALLS   #2, START_SYMBIONT_STREAM        ;
                        58              50  D0 00152          MOVL    R0, STATUS                      ;
                        0C              58  E8 00155          BLBS    STATUS, 18$                     ; 1582
                     11 A3              01  8A 00158          BICB2   #1, 17(SMQ)                     ; 1585
                     11 A3              02  88 0015C          BISB2   #2, 17(SMQ)                     ; 1586
                                        5B  D4 00160          CLRL    NEXT_ACTION                     ; 1587
                                        0B  11 00162          BRB     19$                             ; 1582
                     0C A2              0C  D0 00164 18$:     MOVL    #12, 12(R2)                     ; 1591
                     10 A2              02  88 00168          BISB2   #2, 16(R2)                      ; 1592
                        5B              03  D0 0016C          MOVL    #3, NEXT_ACTION                 ; 1593
                                        54  DD 0016F 19$:     PUSHL   SMQ_N                           ; 1595
                00000000G               EF  01 FB 00171       CALLS   #1, REWRITE_RECORD              ;
                                      00D2  31 00178          BRW     33$                             ; 1473
                        53        20    A2  D0 0017B 20$:     MOVL    32(R2), SMQ_N                   ; 1606
                                        53  DD 0017F          PUSHL   SMQ_N                           ;
                        6A              01  FB 00181          CALLS   #1, READ_RECORD                 ;
                     11 A0              04  8A 00184          BICB2   #4, 17(SMQ)                     ; 1607
                                        50  DD 00188          PUSHL   SMQ                             ; 1608
                                        53  DD 0018A          PUSHL   SMQ_N                           ;
                00000000G               EF  02 FB 0018C       CALLS   #2, STOP_SYMBIONT_STREAM         ;
                                        76  11 00193          BRB     26$                             ; 1609
                        54        20    A2  D0 00195 21$:     MOVL    32(R2), SMQ_N                   ; 1621
```

```
                                 54  DD 00199          PUSHL   SMQ_N
                              6A 01  FB 0019B          CALLS   #1, READ_RECORD
                              53 50  D0 0019E          MOVL    R0, SMQ
                           10 A3 08  8A 001A1          BICB2   #8, 16(SMQ)                    ; 1622
                                 53  DD 001A5          PUSHL   SMQ                            ; 1623
                                 54  DD 001A7          PUSHL   SMQ_N
                 00000000G   EF 02  FB 001A9          CALLS   #2, PAUSE_EXECUTION
                              58 50  D0 001B0          MOVL    R0, STATUS
                              2F 58  E9 001B3          BLBC    STATUS, 23$                    ; 1624
                                 38  11 001B6          BRB     24$                            ; 1625
                           54 20 A2  D0 001B8 22$:      MOVL    32(R2), SMQ_N                  ; 1637
                                 54  DD 001BC          PUSHL   SMQ_N
                              6A 01  FB 001BE          CALLS   #1, READ_RECORD
                              53 50  D0 001C1          MOVL    R0, SMQ
                           10 A3 40  8F 8A 001C4       BICB2   #64, 16(SMQ)                   ; 1638
                                 31  A2 9F 001C9       PUSHAB  49(R2)                         ; 1642
                           7E 30 A2  9A 001CC          MOVZBL  48(R2), -(SP)
                           7E 28 A2  7D 001D0          MOVQ    40(R2), -(SP)
                                 24  A2 DD 001D4       PUSHL   36(R2)
                                 53  DD 001D7          PUSHL   SMQ
                                 54  DD 001D9          PUSHL   SMQ_N
                 00000000G   EF 07  FB 001DB          CALLS   #7, RESUME_EXECUTION
                              58 50  D0 001E2          MOVL    R0, STATUS
                                 53  DD 001E5 23$:     PUSHL   SMQ                            ; 1643
                                 54  DD 001E7          PUSHL   SMQ_N
                 00000000G   EF 02  FB 001E9          CALLS   #2, FIND_PENDING_JOBS
                          FF32 31  31 001F0 24$:      BRW     15$                            ; 1644
                           10 A0 20  8A 001F3 25$:      BICB2   #32, 16(SMQ)                   ; 1656
                              53 20  A2 D0 001F7       MOVL    32(R2), SMQ_N                  ; 1657
                                 53  DD 001FB          PUSHL   SMQ_N
                              6A 01  FB 001FD          CALLS   #1, READ_RECORD
                                 50  DD 00200          PUSHL   SMQ                            ; 1658
                                 53  DD 00202          PUSHL   SMQ_N
                 00000000G   EF 02  FB 00204          CALLS   #2, RESET_EXECUTOR_QUEUE
                                 53  DD 0020B 26$:     PUSHL   SMQ_N                          ; 1659
                 00000000G   EF 01  FB 0020D 27$:     CALLS   #1, REWRITE_RECORD
                                 5B  D4 00214 28$:     CLRL    NEXT_ACTION                    ; 1660
                                 35  11 00216          BRB     33$                            ; 1473
                                 42  A2 9F 00218 29$:     PUSHAB  66(R2)                     ; 1670
                              7E 40  A2 3C 0021B          MOVZWL  64(R2 , -(SP)
                                 20  A2 9F 0021F       PUSHAB  32(R2)                         ; 1668
                           00000000' EF 9F 00222       PUSHAB  THIS_SYSID                     ; 1666
                 00000000G   EF 04  FB 00228          CALLS   #4, BROADCAST_MESSAGE          ; 1670
                                 19  11 0022F          BRB     32$                            ; 1671
                                 20  A2 DD 00231 30$:     PUSHL   32(R2)                     ; 1679
                                 70  A2 9F 00234          PUSHAB  112(R2)                    ; 1678
                 00000000G   EF 02  FB 00237          CALLS   #2, SEND_SERVICE_RESPONSE_MESSAGE
                                 0A  11 0023E          BRB     32$                            ; 1680
                                 20  A2 DD 00240 31$:     PUSHL   32(R2)                     ; 1686
                 00000000G   EF 01  FB 00243          CALLS   #1, DELETE_FILES
                              5B 01  D0 0024A 32$:     MOVL    #1, NEXT_ACTION                ; 1687
                              50 58  D0 0024D 33$:     MOVL    STATUS, R0                     ; 1695
                                 04  00250          RET

; Routine Size:  593 bytes,    Routine Base:  CODE + 0195
```

```
;  660     1696  1 GLOBAL ROUTINE SCAN_INCOMPLETE_SERVICES(EVENT,P1,P2,P3,P4): NOVALUE=
;  661     1697  1
;  662     1698  1 !++
;  663     1699  1 !
;  664     1700  1 ! FUNCTIONAL DESCRIPTION:
;  665     1701  1 !       This routine scans the incomplete services list when a specified event
;  666     1702  1 !       that allows an incomplete service to progress has occurred.
;  667     1703  1 !
;  668     1704  1 ! INPUT PARAMETERS:
;  669     1705  1 !       EVENT           - Code identifying the event.
;  670     1706  1 !       P1-P4           - Event-dependent parameters.
;  671     1707  1 !
;  672     1708  1 ! IMPLICIT INPUTS:
;  673     1709  1 !       NONE
;  674     1710  1 !
;  675     1711  1 ! OUTPUT PARAMETERS:
;  676     1712  1 !       NONE
;  677     1713  1 !
;  678     1714  1 ! IMPLICIT OUTPUTS:
;  679     1715  1 !       NONE
;  680     1716  1 !
;  681     1717  1 ! ROUTINE VALUE:
;  682     1718  1 !       NONE
;  683     1719  1 !
;  684     1720  1 ! SIDE EFFECTS:
;  685     1721  1 !       NONE
;  686     1722  1 !
;  687     1723  1 !--
;  688     1724  1
;  689     1725  2 BEGIN
;  690     1726  2 LOCAL
;  691     1727  2         PRED_MODIFIED,                  ! True if predecessor modified
;  692     1728  2         SRQ_NP,                         ! Record number of predecessor of SRQ
;  693     1729  2         SRQ_P:          REF BBLOCK,     ! Pointer to predecessor of SRQ
;  694     1730  2         SRQ_N;                          ! Record number of SRQ
;  695     1731  2
;  696     1732  2
;  697     1733  2 ! Search the incomplete service list for those that are affected by the
;  698     1734  2 ! specified event and process these.
;  699     1735  2 !
;  700     1736  2 PRED_MODIFIED = FALSE;
;  701     1737  2 SRQ_P = READ_RECORD(SRQ_NP = SQH$K_RECNO);
;  702     1738  2 SRQ_N = .SRQ_P[SQH$L_INCOMPLETE_SERVICE_LIST];
;  703     1739  2 WHILE .SRQ_N NEQ 0 DO
;  704     1740  3     BEGIN
;  705     1741  3     LOCAL
;  706     1742  3         SRQ:            REF BBLOCK,     ! Pointer to SRQ
;  707     1743  3         SRQ_NS,                         ! Record number of successor of SRQ
;  708     1744  3         STATUS,                         ! Request status
;  709     1745  3         NEXT_ACTION;                    ! Code for next action
;  710     1746  3
;  711     1747  3
;  712     1748  3     SRQ = READ_RECORD(.SRQ_N);
;  713     1749  3     SRQ_NS = .SRQ[SYM$L_LINK];
;  714     1750  3
;  715     1751  3
;  716     1752  3     ! Check for corrupted incomplete services list. If an incorrect record type
```

```
;  717          1753  3      ! is found, truncate the list. The remaining records are either already
;  718          1754  3      ! linked to another list, or they will be lost until a cold start operation
;  719          1755  3      ! is performed. Pruning these unwanted records (most likely free list or
;  720          1756  3      ! job header records) from the incomplete services list will prevent
;  721          1757  3      ! reading them every time SCAN_INCOMPLETE_SERVICES is called.
;  722          1758  3      !
;  723          1759  3      IF .SRQ[SYM$B_TYPE] NEQ SYM$K_SRQ
;  724          1760  3      THEN
;  725          1761  4          BEGIN
;  726          1762  4          DIAG_TRACE[12] = .DIAG_TRACE[12] + 1;
;  727          1763  4          DIAG_TRACE[13] = .SRQ[SYM$B_TYPE] * 65536 + .SRQ_N;
;  728          1764  4          IF .FLAGS[FLAGS_V_LOG_QF_REPAIR]
;  729          1765  4              THEN SIGNAL(JBC$_DIAGNOSTIC OR STS$K_INFO, 1,
;  730          1766  4                  $DESCRIPTOR('on-line repair of incomplete services list') );
;  731          1767  4          IF .SRQ_NP EQL SQH$K_RECNO
;  732          1768  4              THEN SRQ_P[SQH$L_INCOMPLETE_SERVICE_LIST] = 0
;  733          1769  4              ELSE SRQ_P[SYM$L_LINK] = 0;
;  734          1770  4          PRED_MODIFIED = TRUE;
;  735          1771  4          EXITLOOP;
;  736          1772  3          END;
;  737          1773  3
;  738          1774  3
;  739          1775  3      STATUS = SS$_NORMAL;
;  740          1776  3      NEXT_ACTION = K_RELEASE;
;  741          1777  3
;  742          1778  3
;  743          1779  3      CASE .EVENT FROM ISRV_K_REMOTE TO ISRV_K_PURGE_SJH OF
;  744          1780  3          SET
;  745          1781  3
;  746          1782  3
;  747          1783  3          [ISRV_K_REMOTE]:
;  748          1784  4              BEGIN
;  749          1785  5              IF SYSID_EQL(THIS_SYSID, SRQ[SRQ$T_RECEIVING_SYSID])
;  750          1786  4              AND NOT .SRQ[SRQ$V_STALLED]
;  751          1787  4              THEN
;  752          1788  4                  STATUS = PROCESS_REMOTE_SERVICES(.SRQ; NEXT_ACTION);
;  753          1789  3              END;
;  754          1790  3
;  755          1791  3
;  756          1792  3          [ISRV_K_SYNCHRONIZE]:
;  757          1793  4              BEGIN
;  758          1794  4              BIND
;  759          1795  4                  SJH_N           = P1,                   ! Record number of SJH
;  760          1796  4                  STS             = P2;                   ! Completion status
;  761          1797  4
;  762          1798  4              IF .SRQ[SRQ$L_FUNCTION_CODE] EQL SRQ$K_SYNCHRONIZE_JOB
;  763          1799  4              AND .SRQ[SRQ$L_P1] EQL .SJH_N
;  764          1800  4              THEN
;  765          1801  5                  BEGIN
;  766          1802  5                  NEXT_ACTION = K_COMPLETE;
;  767          1803  5                  STATUS = .STS;
;  768          1804  4                  END;
;  769          1805  3              END;
;  770          1806  3
;  771          1807  3
;  772          1808  3          [ISRV_K_SYMBIONT]:
;  773          1809  4              BEGIN
```

```
;  774          1810  4              BIND
;  775          1811  4                  SMQ_N           = P1,                   ! Record number of SMQ
;  776          1812  4                  SMQ             = P2,                   ! Pointer to SMQ
;  777          1813  4                  FUNC            = P3,                   ! Function completed
;  778          1814  4                  STS             = P4;                   ! Completion status
;  779          1815  4
;  780          1816  4              IF .SRQ[SRQ$L_FUNCTION_CODE] EQL SRQ$K_START_SYMBIONT
;  781          1817  4              AND .SRQ[SRQ$L_P1] EQL .SMQ_N
;  782          1818  5              AND (.FUNC EQL 0 OR .FUNC EQL .SRQ[SRQ$L_FUNCTION_CODE])
;  783          1819  4              THEN
;  784          1820  5                  BEGIN
;  785          1821  5                  NEXT_ACTION = K_COMPLETE;
;  786          1822  5                  STATUS = .STS;
;  787          1823  4                  END;
;  788          1824  3              END;
;  789          1825  3
;  790          1826  3
;  791          1827  3          [ISRV_K_PURGE_SYSID]:
;  792          1828  4              BEGIN
;  793          1829  4              BIND
;  794          1830  4                  SYSID           = P1;                   ! Pointer to system ID
;  795          1831  4
;  796          1832  5              IF SYSID_EQL(.SYSID, SRQ[SRQ$T_SENDING_SYSID])
;  797          1833  4              THEN
;  798          1834  4                  NEXT_ACTION = K_DEALLOCATE
;  799          1835  4
;  800          1836  5              ELSE IF SYSID_EQL(.SYSID, SRQ[SRQ$T_RECEIVING_SYSID])
;  801          1837  4              THEN
;  802          1838  5                  BEGIN
;  803          1839  5                  STATUS = JBC$_SYSFAIL OR STS$K_ERROR;
;  804          1840  5                  NEXT_ACTION = K_COMPLETE;
;  805          1841  4                  END;
;  806          1842  3              END;
;  807          1843  3
;  808          1844  3
;  809          1845  3          [ISRV_K_PURGE_SMQ]:
;  810          1846  4              BEGIN
;  811          1847  4              BIND
;  812          1848  4                  SMQ_N           = P1;                   ! Record number of SMQ
;  813          1849  4
;  814          1850  4              IF
;  815        P 1851  4                  ONEOF (.SRQ[SRQ$L_FUNCTION_CODE], BMSK_(
;  816        P 1852  4                      SRQ$K_START_QUEUE,
;  817        P 1853  4                      SRQ$K_STOP_QUEUE,
;  818        P 1854  4                      SRQ$K_PAUSE_QUEUE,
;  819        P 1855  4                      SRQ$K_RESUME_QUEUE,
;  820        P 1856  4                      SRQ$K_RESET_QUEUE,
;  821          1857  5                      SRQ$K_START_SYMBIONT))
;  822          1858  4                  AND .SRQ[SRQ$L_P1] EQL .SMQ_N
;  823          1859  4              THEN
;  824          1860  4                  NEXT_ACTION = K_COMPLETE;
;  825          1861  3              END;
;  826          1862  3
;  827          1863  3
;  828          1864  3          [ISRV_K_PURGE_SJH]:
;  829          1865  4              BEGIN
;  830          1866  4              BIND
```

```
;  831          1867  4                  SJH_N           = P1;                    ! Record number of SJH
;  832          1868  4
;  833          1869  4              IF
;  834        P 1870  4                  ONEOF (.SRQ[SRQ$L_FUNCTION_CODE], BMSK_(
;  835        P 1871  4                      SRQ$K_START_JOB,
;  836          1872  5                      SRQ$K_ABORT_JOB))
;  837          1873  4                  AND .SRQ[SRQ$L_P2] EQL .SJH_N
;  838          1874  4              THEN
;  839          1875  4                  NEXT_ACTION = K_COMPLETE;
;  840          1876  3              END;
;  841          1877  3
;  842          1878  3
;  843          1879  3          TES;
;  844          1880  3
;  845          1881  3
;  846          1882  3      IF .NEXT_ACTION EQL K_COMPLETE
;  847          1883  3      THEN
;  848          1884  4          BEGIN
;  849          1885  4
;  850          1886  4          ! If no response is required, merely deallocate the SRQ.                    ;
;  851          1887  4          !
;  852          1888  4          IF .SRQ[SRQ$V_NO_RESPONSE]
;  853          1889  4          THEN
;  854          1890  4              NEXT_ACTION = K_DEALLOCATE
;  855          1891  4
;  856          1892  4
;  857          1893  4          ! If the response can be sent locally, send it and deallocate the SRQ.
;  858          1894  4          !
;  859          1895  5          ELSE IF SYSID_EQL(THIS_SYSID, SRQ[SRQ$T_SENDING_SYSID])
;  860          1896  4          THEN
;  861          1897  5              BEGIN
;  862          1898  5              SEND_SERVICE_RESPONSE_MESSAGE(SRQ[SRQ$T_SRB], .STATUS);
;  863          1899  5              NEXT_ACTION = K_DEALLOCATE;
;  864          1900  5              END
;  865          1901  5
;  866          1902  5
;  867          1903  5          ! Otherwise, convert the SRQ to a "response" request and forward it
;  868          1904  5          ! to the sending job controller.
;  869          1905  5          !
;  870          1906  4          ELSE
;  871          1907  5              BEGIN
;  872          1908  5              COPY_SYSID(SRQ[SRQ$T_SENDING_SYSID], SRQ[SRQ$T_RECEIVING_SYSID]);
;  873          1909  5              COPY_SYSID(THIS_SYSID, SRQ[SRQ$T_SENDING_SYSID]);
;  874          1910  5              SRQ[SRQ$L_FUNCTION_CODE] = SRQ$K_RESPONSE;
;  875          1911  5              SRQ[SRQ$L_P1] = .STATUS;
;  876          1912  5              SRQ[SRQ$V_STALLED] = FALSE;
;  877          1913  5              ENTER_REMOTE_REQUEST(SRQ[SRQ$T_RECEIVING_SYSID]);
;  878          1914  5              NEXT_ACTION = K_REWRITE;
;  879          1915  4              END;
;  880          1916  3          END;
;  881          1917  3
;  882          1918  3
;  883          1919  3      CASE .NEXT_ACTION FROM K_DEALLOCATE TO K_REWRITE OF
;  884          1920  3          SET
;  885          1921  3
;  886          1922  3
;  887          1923  3          [K_DEALLOCATE]:
```

```
;  888          1924  4              BEGIN
;  889          1925  4              IF .SRQ_NP EQL SQH$K_RECNO
;  890          1926  4                  THEN SRQ_P[SQH$L_INCOMPLETE_SERVICE_LIST] = .SRQ_NS
;  891          1927  4                  ELSE SRQ_P[SYM$L_LINK] = .SRQ_NS;
;  892          1928  4
;  893          1929  4              ! First, rewrite the predecessor, then deallocate the SRQ.
;  894          1930  4              ! If done in the opposite order, a crash after the deallocate
;  895          1931  4              ! can result in a corrupted INCOMPLETE_SERVICE_LIST, which
;  896          1932  4              ! will then result in a queue format error on warm/cold start.
;  897          1933  4              !
;  898          1934  4              FLUSH_RECORD(.SRQ_NP);
;  899          1935  4              DEALLOCATE_RECORD(.SRQ_N);
;  900          1936  3              END;
;  901          1937  3
;  902          1938  3
;  903          1939  3          [K_RELEASE]:
;  904          1940  4              BEGIN
;  905          1941  4              IF TESTBITSC(PRED_MODIFIED)
;  906          1942  4                  THEN REWRITE_RECORD(.SRQ_NP)
;  907          1943  4                  ELSE RELEASE_RECORD(.SRQ_NP);
;  908          1944  4              SRQ_NP = .SRQ_N;
;  909          1945  4              SRQ_P = .SRQ;
;  910          1946  3              END;
;  911          1947  3
;  912          1948  3
;  913          1949  3          [K_REWRITE]:
;  914          1950  4              BEGIN
;  915          1951  4              IF TESTBITSS(PRED_MODIFIED)
;  916          1952  4                  THEN REWRITE_RECORD(.SRQ_NP)
;  917          1953  4                  ELSE RELEASE_RECORD(.SRQ_NP);
;  918          1954  4              SRQ_NP = .SRQ_N;
;  919          1955  4              SRQ_P = .SRQ;
;  920          1956  3              END;
;  921          1957  3
;  922          1958  3
;  923          1959  3              TES;
;  924          1960  3
;  925          1961  3
;  926          1962  3          SRQ_N = .SRQ_NS;
;  927          1963  2          END;
;  928          1964  2
;  929          1965  2
;  930          1966  2 IF .PRED_MODIFIED
;  931          1967  2     THEN REWRITE_RECORD(.SRQ_NP)
;  932          1968  2     ELSE RELEASE_RECORD(.SRQ_NP);
;  933          1969  1 END;


20 72 69 61 70 65 72 20 65 6E 69 6C 2D 6E 6F  003E6 P.AAB:  .ASCII  \on-line repair of incomplete services li\  ;
73 20 65 74 65 6C 70 6D 6F 63 6E 69 20 66 6F  003F5                                                          ;
                  69 6C 20 73 65 63 69 76 72 65  00404                                                          ;
                                                 74 73  0040E          .ASCII  \ `\                           ;
                                          0000002A  00410 P.AAA:  .LONG   42                                   ;
                                          00000000'  00414        .ADDRESS P.AAB                               ;
```

```
                                                 0FFC 00000          .ENTRY  SCAN_INCOMPLETE_SERVICES, Save R2,R3,R4,R5,-; 1696
                                                                             R6,R7,R8,R9,R10,R11
                                              5A 00000000G   EF  9E 00002    MOVAB   REWRITE_RECORD, R10                  ;
                                              59 00000000'   EF  9E 00009    MOVAB   THIS_SYSID, R9                      ;
                                                             57  D4 00010    CLRL    PRED_MODIFIED                       ; 1736
                                              55             01  D0 00012    MOVL    #1, SRQ_NP                          ; 1737
                                                             01  DD 00015    PUSHL   #1                                  ;
                                    00000000G EF             01  FB 00017    CALLS   #1, READ_RECORD                     ;
                                              53             50  D0 0001E    MOVL    R0, SRQ_P                           ;
                                              56       44    A3  D0 00021    MOVL    68(SRQ_P), SRQ_N                    ; 1738
                                                             4D  13 00025 1$: BEQL    5$                                  ; 1739
                                                             56  DD 00027    PUSHL   SRQ_N                               ; 1748
                                    00000000G EF             01  FB 00029    CALLS   #1, READ_RECORD                     ;
                                              52             50  D0 00030    MOVL    R0, SRQ                             ;
                                              58             62  D0 00033    MOVL    (SRQ), SRQ_NS                       ; 1749
                                              09       04    A2  91 00036    CMPB    4(SRQ), #9                          ; 1759
                                                             3B  13 0003A    BEQL    6$                                  ;
                                                   FE28      C9  D6 0003C    INCL    DIAG_TRACE+48                       ; 1762
                                              50       04    A2  9A 00040    MOVZBL  4(SRQ), R0                          ; 1763
                                        50    50             10  78 00044    ASHL    #16, R0, R0                         ;
                                   FE2C C9    50             56  C1 00048    ADDL3   SRQ_N, R0, DIAG_TRACE+52            ;
                                        12       FA    A9    05  E1 0004E    BBC     #5, FLAGS+2, 2$                     ; 1764
                                                       A2    AF  9F 00053    PUSHAB  P.AAA                               ; 1766
                                                             01  DD 00056    PUSHL   #1                                  ; 1765
                                                 000484B3    8F  DD 00058    PUSHL   #296115                             ;
                                    00000000G 00             03  FB 0005E    CALLS   #3, LIB$SIGNAL                      ;
                                              01             55  D1 00065 2$: CMPL    SRQ_NP, #1                          ; 1767
                                                             05  12 00068    BNEQ    3$                                  ;
                                                       44    A3  D4 0006A    CLRL    68(SRQ_P)                           ; 1768
                                                             02  11 0006D    BRB     4$                                  ;
                                                             63  D4 0006F 3$: CLRL    (SRQ_P)                             ; 1769
                                              57             01  D0 00071 4$: MOVL    #1, PRED_MODIFIED                   ; 1770
                                                   015F      31  00074 5$:   BRW     34$                                 ; 1761
                                              54             01  D0 00077 6$: MOVL    #1, STATUS                          ; 1775
                                              5B             02  D0 0007A    MOVL    #2, NEXT_ACTION                     ; 1776
                                        05    00       04    AC  CF 0007D    CASEL   EVENT, #0, #5                       ; 1779
                                  0060  003F       002A        000C    00082 7$:   .WORD   8$-7$,-                         ;
                                                   009E        008C    0008A       9$-7$,-                                 ;
                                                                               10$-7$,-                                ;
                                                                               12$-7$,-                                ;
                                                                               16$-7$,-                                ;
                                                                               17$-7$                                  ;
                                                 1A    A2          69  D1 0008E 8$: CMPL    THIS_SYSID, 26(SRQ)         ; 1785
                                                             6F  12 00092    BNEQ    15$                                 ;
                                                 1E    A2    04    A9  B1 00094    CMPW    THIS_SYSID+4, 30(SRQ)       ;
                                                             68  12 00099    BNEQ    15$                                 ;
                                              56 10    A2          01  E0 0009B    BBS     #1, 16(SRQ), 13$            ; 1786
                                                             52  DD 000A0    PUSHL   SRQ                                 ; 1788
                                                 FCD6    CF          01  FB 000A2    CALLS   #1, PROCESS_REMOTE_SERVICES  ;
                                                             54  D0 000A7    MOVL    R0, STATUS                          ;
                                                             4A  11 000AA    BRB     13$                                 ; 1779
                                                       03    0C  A2  D1 000AC 9$: CMPL    12(SRQ), #3                 ; 1798
                                                             7E  12 000B0    BNEQ    18$                                 ;
                                                 08    AC    20  A2  D1 000B2    CMPL    32(SRQ), SJH_N              ; 1799
                                                             7B  12 000B7    BNEQ    20$                                 ;
                                                             5B  D4 000B9    CLRL    NEXT_ACTION                         ; 1802
```

```
                    54        0C   AC  D0 000BB          MOVL    STS, STATUS                             ; 1803
                              0C   73  11 000BF          BRB     20$                                     ; 1779
                              0C   A2  D1 000C1 10$:     CMPL    12(SRQ), #12                            ; 1816
                                   6D  12 000C5          BNEQ    20$
                          08   AC   20  A2  D1 000C7      CMPL    32(SRQ), SMQ_N                          ; 1817
                                   66  12 000CC          BNEQ    20$
                               10  AC   D5 000CE         TSTL    FUNC                                    ; 1818
                                   07  13 000D1          BEQL    11$
                          0C   A2   10  AC  D1 000D3      CMPL    FUNC, 12(SRQ)
                                   5A  12 000D8          BNEQ    20$
                                   5B  D4 000DA 11$:     CLRL    NEXT_ACTION                             ; 1821
                               14  A2   AC  D0 000DC      MOVL    STS, STATUS                             ; 1822
                                   52  11 000E0          BRB     20$                                     ; 1779
                          50       08   AC  D0 000E2 12$:     MOVL    SYSID, R0                           ; 1832
                          14   A2       60  D1 000E6      CMPL    (R0), 20(SRQ)
                                   0C  12 000EA          BNEQ    14$
                          18   A2   04   A0  B1 000EC      CMPW    4(R0), 24(SRQ)
                                   05  12 000F1          BNEQ    14$
                               5B       01  D0 000F3      MOVL    #1, NEXT_ACTION                         ; 1834
                                   3C  11 000F6 13$:     BRB     20$
                          1A   A2       60  D1 000F8 14$:     CMPL    (R0), 26(SRQ)                       ; 1836
                                   36  12 000FC          BNEQ    20$
                          1E   A2   04   A0  B1 000FE      CMPW    4(R0), 30(SRQ)
                                   2F  12 00103 15$:     BNEQ    20$
                               54  000480F2  8F  D0 00105      MOVL    #295154, STATUS                     ; 1839
                                   24  11 0010C          BRB     19$                                     ; 1840
                    50 0F880000    8F   0C   A2  78 0010E 16$:     ASHL    12(SRQ), #260571136, R0         ; 1857
                                   1B  18 00117          BGEQ    20$
                          08   AC   20  A2  D1 00119      CMPL    32(SRQ), SMQ_N                          ; 1858
                                   10  11 0011E          BRB     18$
                    50 60000000    8F   0C   A2  78 00120 17$:     ASHL    12(SRQ), #1610612736, R0        ; 1872
                                   09  18 00129          BGEQ    20$
                          08   AC   24  A2  D1 0012B      CMPL    36(SRQ), SJH_N                          ; 1873
                                   02  12 00130 18$:     BNEQ    20$
                                   5B  D4 00132 19$:     CLRL    NEXT_ACTION                             ; 1875
                                   5B  D5 00134 20$:     TSTL    NEXT_ACTION                             ; 1882
                                   4C  12 00136          BNEQ    23$
                               19  10   A2  E8 00138      BLBS    16(SRQ), 21$                            ; 1888
                          14   A2       69  D1 0013C      CMPL    THIS_SYSID, 20(SRQ)                     ; 1895
                                   18  12 00140          BNEQ    22$
                          18   A2   04   A9  B1 00142      CMPW    THIS_SYSID+4, 24(SRQ)
                                   11  12 00147          BNEQ    22$
                                   54  DD 00149          PUSHL   STATUS                                  ; 1898
                               70  A2   9F  00149A         PUSHAB  112(SRQ)
                    00000000G  EF       02  FB 0014E      CALLS   #2, SEND_SERVICE_RESPONSE_MESSAGE
                               5B       01  D0 00155 21$:     MOVL    #1, NEXT_ACTION                     ; 1899
                                   2A  11 00158          BRB     23$                                     ; 1895
                          1A   A2   14   A2  D0 0015A 22$:     MOVL    20(SRQ), 26(SRQ)                    ; 1908
                          1E   A2   18   A2  B0 0015F      MOVW    24(SRQ), 30(SRQ)
                          14   A2       69  D0 00164      MOVL    THIS_SYSID, 20(SRQ)                     ; 1909
                          18   A2   04   A9  B0 00168      MOVW    THIS_SYSID+4, 24(SRQ)
                          0C   A2       0A  D0 0016D      MOVL    #10, 12(SRQ)                            ; 1910
                          20   A2       54  D0 00171      MOVL    STATUS, 32(SRQ)                         ; 1911
                          10   A2       02  8A 00175      BICB2   #2, 16(SRQ)                             ; 1912
                               1A   A2  9F  00179         PUSHAB  26(SRQ)                                 ; 1913
                    0000V      CF       01  FB 0017C      CALLS   #1, ENTER_REMOTE_REQUEST
                               5B       03  D0 00181      MOVL    #3, NEXT_ACTION                         ; 1914
```

```
                02          01          5B  CF 00184 23$:    CASEL   NEXT_ACTION, #1, #2                      ; 1919
              002E        0028        0006     00188 24$:    .WORD   25$-24$,-                                ;
                                                               28$-24$,-                                      ;
                                                               29$-24$                                        ;
                            01          55  D1 0018E 25$:    CMPL    SRQ_NP, #1                              ; 1925
                                        06  12 00191         BNEQ    26$                                     ;
                        44  A3          58  D0 00193         MOVL    SRQ_NS, 68(SRQ_P)                       ; 1926
                                        03  11 00197         BRB     27$                                     ;
                            63          58  D0 00199 26$:    MOVL    SRQ_NS, (SRQ_P)                         ; 1927
                                        55  DD 0019C 27$:    PUSHL   SRQ_NP                                  ; 1934
                00000000G   EF          01  FB 0019E         CALLS   #1, FLUSH_RECORD                        ;
                                        56  DD 001A5         PUSHL   SRQ_N                                   ; 1935
                00000000G   EF          01  FB 001A7         CALLS   #1, DEALLOCATE_RECORD                   ;
                                        20  11 001AE         BRB     33$                                     ; 1919
                0D          57          00  E5 001B0 28$:    BBCC    #0, PRED_MODIFIED, 31$                  ; 1941
                                        04  11 001B4         BRB     30$                                     ; 1942
                07          57          00  E3 001B6 29$:    BBCS    #0, PRED_MODIFIED, 31$                  ; 1951
                                        55  DD 001BA 30$:    PUSHL   SRQ_NP                                  ; 1952
                            6A          01  FB 001BC         CALLS   #1, REWRITE_RECORD                      ;
                                        09  11 001BF         BRB     32$                                     ;
                                        55  DD 001C1 31$:    PUSHL   SRQ_NP                                  ; 1953
                00000000G   EF          01  FB 001C3         CALLS   #1, RELEASE_RECORD                      ;
                            55          56  D0 001CA 32$:    MOVL    SRQ_N, SRQ_NP                           ; 1954
                            53          52  D0 001CD         MOVL    SRQ, SRQ_P                              ; 1955
                            56          58  D0 001D0 33$:    MOVL    SRQ_NS, SRQ_N                           ; 1962
                                      FE4F  31 001D3         BRW     1$                                      ; 1739
                            06          57  E9 001D6 34$:    BLBC    PRED_MODIFIED, 35$                      ; 1966
                                        55  DD 001D9         PUSHL   SRQ_NP                                  ; 1967
                            6A          01  FB 001DB         CALLS   #1, REWRITE_RECORD                      ;
                                            04 001DE         RET                                             ;
                                        55  DD 001DF 35$:    PUSHL   SRQ_NP                                  ; 1968
                00000000G   EF          01  FB 001E1         CALLS   #1, RELEASE_RECORD                      ;
                                            04 001E8         RET                                             ; 1969

; Routine Size:  489 bytes,    Routine Base:  CODE + 0418                                                     ;
```

```
;  935          1970  1 GLOBAL ROUTINE REMOTE_BLOCKING_AST: NOVALUE=
;  936          1971  1
;  937          1972  1 !++
;  938          1973  1 !
;  939          1974  1 ! FUNCTIONAL DESCRIPTION:
;  940          1975  1 !       This routine is the blocking AST routine for the job controller remote
;  941          1976  1 !       request lock.  This routine is entered when another job controller
;  942          1977  1 !       attempts to obtain this job controller's remote request lock.
;  943          1978  1 !
;  944          1979  1 ! INPUT PARAMETERS:
;  945          1980  1 !       Standard AST routine parameters (not used).
;  946          1981  1 !
;  947          1982  1 ! IMPLICIT INPUTS:
;  948          1983  1 !       NONE
;  949          1984  1 !
;  950          1985  1 ! OUTPUT PARAMETERS:
;  951          1986  1 !       NONE
;  952          1987  1 !
;  953          1988  1 ! IMPLICIT OUTPUTS:
;  954          1989  1 !       NONE
;  955          1990  1 !
;  956          1991  1 ! ROUTINE VALUE:
;  957          1992  1 !       NONE
;  958          1993  1 !
;  959          1994  1 ! SIDE EFFECTS:
;  960          1995  1 !       NONE
;  961          1996  1 !
;  962          1997  1 !--
;  963          1998  1
;  964          1999  2 BEGIN
;  965          2000  2 LOCAL
;  966          2001  2         STATUS_1,                       ! Status return
;  967          2002  2         STATUS_2;                       ! Status return
;  968          2003  2
;  969          2004  2
;  970          2005  2 ! Convert the lock to null mode to allow the process that has requested the
;  971          2006  2 ! lock to obtain it.
;  972          2007  2 !
;  973        P 2008  2 STATUS_1 = $ENQW(
;  974        P 2009  2     EFN=JBC$K_SYNC_EFN,
;  975        P 2010  2     LKMODE=LCK$K_NLMODE,
;  976        P 2011  2     LKSB=REMOTE_REQUEST_LKSB,
;  977          2012  2     FLAGS=LCK$M_CONVERT);
;  978          2013  2 IF .STATUS_1 THEN STATUS_1 = .REMOTE_REQUEST_LKSB[0];
;  979          2014  2 IF NOT .STATUS_1
;  980          2015  2 THEN
;  981          2016  2     SIGNAL(JBC$_COMREMJBC OR STS$K_ERROR, 0, .STATUS_1);
;  982          2017  2
;  983          2018  2
;  984          2019  2 ! Reconvert the lock to exclusive mode to reenable the blocking AST.
;  985          2020  2 !
;  986        P 2021  2 STATUS_2 = $ENQ(
;  987        P 2022  2     LKMODE=LCK$K_EXMODE,
;  988        P 2023  2     LKSB=REMOTE_REQUEST_LKSB,
;  989        P 2024  2     FLAGS=LCK$M_CONVERT OR LCK$M_NODLCKBLK,
;  990        P 2025  2     ASTADR=REMOTE_COMPLETION_AST,
;  991          2026  2     BLKAST=REMOTE_BLOCKING_AST);
```

```
;  992          2027  2 IF NOT .STATUS_2
;  993          2028  2 THEN
;  994          2029  2     SIGNAL(JBC$_COMREMJBC OR STS$K_ERROR, 0, .STATUS_2);
;  995          2030  1 END;

                                                        .EXTRN  SYS$ENQW, SYS$ENQ

                                                 000C 00000     .ENTRY  REMOTE_BLOCKING_AST, Save R2,R3         ; 1970
                                53 00000000G   00   9E 00002    MOVAB   LIB$SIGNAL, R3
                                52 00000000'   EF   9E 00009    MOVAB   REMOTE_REQUEST_LKSB, R2
                                               7E   7C 00010    CLRQ    -(SP)                                   ; 2012
                                               7E   7C 00012    CLRQ    -(SP)
                                               7E   7C 00014    CLRQ    -(SP)
                                7E                  02   7D 00016    MOVQ    #2, -(SP)
                                                    52   DD 00019    PUSHL   R2
                                7E                  01   7D 0001B    MOVQ    #1, -(SP)
                     00000000G  00                  0B   FB 0001E    CALLS   #11, SYS$ENQW
                                06                  50   E9 00025    BLBC    STATUS_1, 1$                    ; 2013
                                50                  62   3C 00028    MOVZWL  REMOTE_REQUEST_LKSB, STATUS_1
                                0D                  50   E8 0002B    BLBS    STATUS_1, 2$                    ; 2014
                                                    50   DD 0002E 1$: PUSHL   STATUS_1                        ; 2016
                                                    7E   D4 00030    CLRL    -(SP)
                                   00048412         8F   DD 00032    PUSHL   #295954
                                63                  03   FB 00038    CALLS   #3, LIB$SIGNAL
                                                    7E   7C 0003B 2$: CLRQ    -(SP)                           ; 2026
                                         C0         AF   9F 0003D    PUSHAB  REMOTE_BLOCKING_AST
                                                    7E   D4 00040    CLRL    -(SP)
                                      0000V         CF   9F 00042    PUSHAB  REMOTE_COMPLETION_AST
                                                    7E   7C 00046    CLRQ    -(SP)
                                7E          0402    8F   3C 00048    MOVZWL  #1026, -(SP)
                                                    52   DD 0004D    PUSHL   R2
                                                    05   DD 0004F    PUSHL   #5
                                                    7E   D4 00051    CLRL    -(SP)
                     00000000G  00                  0B   FB 00053    CALLS   #11, SYS$ENQ
                                0D                  50   E8 0005A    BLBS    STATUS_2, 3$                    ; 2027
                                                    50   DD 0005D    PUSHL   STATUS_2                        ; 2029
                                                    7E   D4 0005F    CLRL    -(SP)
                                   00048412         8F   DD 00061    PUSHL   #295954
                                63                  03   FB 00067    CALLS   #3, LIB$SIGNAL
                                                         04 0006A 3$: RET                                     ; 2030

; Routine Size:  107 bytes,    Routine Base:  CODE + 0601
```

```
;  997          2031  1 ROUTINE REMOTE_COMPLETION_NONAST: NOVALUE=
;  998          2032  1
;  999          2033  1 !++
; 1000          2034  1 !
; 1001          2035  1 ! FUNCTIONAL DESCRIPTION:
; 1002          2036  1 !       This routine is scheduled to execute by the completion AST routine for
; 1003          2037  1 !       reconversion of the job controller remote request lock to exclusive
; 1004          2038  1 !       mode, which is entered when another job controller has obtained and
; 1005          2039  1 !       released this job controller's remote request lock.
; 1006          2040  1 !
; 1007          2041  1 ! INPUT PARAMETERS:
; 1008          2042  1 !       NONE
; 1009          2043  1 !
; 1010          2044  1 ! IMPLICIT INPUTS:
; 1011          2045  1 !       NONE
; 1012          2046  1 !
; 1013          2047  1 ! OUTPUT PARAMETERS:
; 1014          2048  1 !       NONE
; 1015          2049  1 !
; 1016          2050  1 ! IMPLICIT OUTPUTS:
; 1017          2051  1 !       NONE
; 1018          2052  1 !
; 1019          2053  1 ! ROUTINE VALUE:
; 1020          2054  1 !       NONE
; 1021          2055  1 !
; 1022          2056  1 ! SIDE EFFECTS:
; 1023          2057  1 !       NONE
; 1024          2058  1 !
; 1025          2059  1 !--
; 1026          2060  1
; 1027          2061  2 BEGIN
; 1028          2062  2
; 1029          2063  2 ! Get the current time.
; 1030          2064  2 !
; 1031          2065  2 $GETTIM(TIMADR=CUR_TIME);
; 1032          2066  2
; 1033          2067  2
; 1034          2068  2 IF .QUEUE_FAB[FAB$W_IFI] NEQ 0
; 1035          2069  2 THEN
; 1036          2070  3     BEGIN
; 1037          2071  3
; 1038          2072  3     ! Lock the queue file.
; 1039          2073  3     !
; 1040          2074  3     LOCK_QUEUE_FILE();
; 1041          2075  3
; 1042          2076  3
; 1043          2077  3     ! Search the incomplete services list to perform actions requested by the
; 1044          2078  3     ! remote job controller.
; 1045          2079  3     !
; 1046          2080  3     SCAN_INCOMPLETE_SERVICES(ISRV_K_REMOTE);
; 1047          2081  3
; 1048          2082  3
; 1049          2083  3     ! Unlock the queue file.
; 1050          2084  3     !
; 1051          2085  3     UNLOCK_QUEUE_FILE();
; 1052          2086  2     END;
; 1053          2087  1 END;
```

```
                                                        .EXTRN  SYS$GETTIM

                                                0000 00000 REMOTE_COMPLETION_NONAST:
                                                           .WORD   Save nothing                                 ; 2031
                          00000000'  EF   9F 00002         PUSHAB  CUR_TIME                                    ; 2065
00000000G   00                       01   FB 00008         CALLS   #1, SYS$GETTIM
                          00000000'  EF   B5 0000F         TSTW    QUEUE_FAB+2                                 ; 2068
                                     15   13 00015         BEQL    1$
00000000G   EF                       00   FB 00017         CALLS   #0, LOCK_QUEUE_FILE                         ; 2074
                                     7E   D4 0001E         CLRL    -(SP)                                       ; 2080
    FD87    CF                       01   FB 00020         CALLS   #1, SCAN_INCOMPLETE_SERVICES
00000000G   EF                       00   FB 00025         CALLS   #0, UNLOCK_QUEUE_FILE                       ; 2085
                                     04   0002C 1$:        RET                                                 ; 2087

; Routine Size:  45 bytes,    Routine Base:  CODE + 066C
```

```
;  1055          2088  1 ROUTINE REMOTE_COMPLETION_AST: NOVALUE=
;  1056          2089  1
;  1057          2090  1 !++
;  1058          2091  1 !
;  1059          2092  1 ! FUNCTIONAL DESCRIPTION:
;  1060          2093  1 !       This routine is the completion AST routine for reconversion of the job
;  1061          2094  1 !       controller remote request lock to exclusive mode.  This routine is
;  1062          2095  1 !       entered when another job controller has obtained and released this job
;  1063          2096  1 !       controller's remote request lock.
;  1064          2097  1 !
;  1065          2098  1 ! INPUT PARAMETERS:
;  1066          2099  1 !       Standard AST routine parameters (not used).
;  1067          2100  1 !
;  1068          2101  1 ! IMPLICIT INPUTS:
;  1069          2102  1 !       NONE
;  1070          2103  1 !
;  1071          2104  1 ! OUTPUT PARAMETERS:
;  1072          2105  1 !       NONE
;  1073          2106  1 !
;  1074          2107  1 ! IMPLICIT OUTPUTS:
;  1075          2108  1 !       NONE
;  1076          2109  1 !
;  1077          2110  1 ! ROUTINE VALUE:
;  1078          2111  1 !       NONE
;  1079          2112  1 !
;  1080          2113  1 ! SIDE EFFECTS:
;  1081          2114  1 !       NONE
;  1082          2115  1 !
;  1083          2116  1 !--
;  1084          2117  1
;  1085          2118  2 BEGIN
;  1086          2119  2
;  1087          2120  2 ! Check status of the $ENQ.
;  1088          2121  2 !
;  1089          2122  2 IF NOT .REMOTE_REQUEST_LKSB[0]
;  1090          2123  2 THEN
;  1091          2124  2     SIGNAL(JBC$_COMREMJBC OR STS$K_ERROR, 0, .REMOTE_REQUEST_LKSB[0]);
;  1092          2125  2
;  1093          2126  2
;  1094          2127  2 ! Schedule the companion routine to execute.
;  1095          2128  2 !
;  1096          2129  2 SCHEDULE_NONAST(REMOTE_COMPLETION_NONAST);
;  1097          2130  1 END;


                                0004 00000 REMOTE_COMPLETION_AST:
                                                .WORD   Save R2                                         ; 2088
                            52 00000000'    EF  9E 00002            MOVAB   REMOTE_REQUEST_LKSB, R2
                            12              62  E8 00009            BLBS    REMOTE_REQUEST_LKSB, 1$     ; 2122
                            7E              62  3C 0000C            MOVZWL  REMOTE_REQUEST_LKSB, -(SP)  ; 2124
                                            7E  D4 0000F            CLRL    -(SP)
                               00048412     8F  DD 00011            PUSHL   #295954
                    00000000G   00          03  FB 00017            CALLS   #3, LIB$SIGNAL
                                        B2  AF  9F 0001E 1$:        PUSHAB  REMOTE_COMPLETION_NONAST        ; 2129
```

```
                    00000000G  EF          01  FB 00021          CALLS   #1, SCHEDULE_NONAST                    ;
                                               04 00028          RET                                           ; 2130

; Routine Size:  41 bytes,    Routine Base:  CODE + 0699
```

```
;  1099          2131  1 ROUTINE ENTER_REMOTE_REQUEST(SYSID): NOVALUE=
;  1100          2132  1
;  1101          2133  1 !++
;  1102          2134  1 !
;  1103          2135  1 ! FUNCTIONAL DESCRIPTION:
;  1104          2136  1 !       This routine requests services of another job controller.
;  1105          2137  1 !
;  1106          2138  1 ! INPUT PARAMETERS:
;  1107          2139  1 !       SYSID           - Address of the system ID of the target.
;  1108          2140  1 !
;  1109          2141  1 ! IMPLICIT INPUTS:
;  1110          2142  1 !       NONE
;  1111          2143  1 !
;  1112          2144  1 ! OUTPUT PARAMETERS:
;  1113          2145  1 !       NONE
;  1114          2146  1 !
;  1115          2147  1 ! IMPLICIT OUTPUTS:
;  1116          2148  1 !       NONE
;  1117          2149  1 !
;  1118          2150  1 ! ROUTINE VALUE:
;  1119          2151  1 !       NONE
;  1120          2152  1 !
;  1121          2153  1 ! SIDE EFFECTS:
;  1122          2154  1 !       NONE
;  1123          2155  1 !
;  1124          2156  1 !--
;  1125          2157  1
;  1126          2158  2 BEGIN
;  1127          2159  2 MAP
;  1128          2160  2         SYSID:          REF BBLOCK;     ! Pointer to system ID
;  1129          2161  2 LOCAL
;  1130          2162  2         LKSB:           REF BBLOCK,     ! Pointer to LKSB from dynamic memory
;  1131          2163  2         RESNAM_DESC:    VECTOR[2],      ! Descriptor for resource name
;  1132          2164  2         RESNAM:         BBLOCK[10],     ! Buffer for resource name
;  1133          2165  2         STATUS;                         ! Status return
;  1134          2166  2
;  1135          2167  2
;  1136          2168  2 ! Allocate and initialize the LKSB.
;  1137          2169  2 !
;  1138          2170  2 LKSB = ALLOCATE_MEMORY();
;  1139          2171  2
;  1140          2172  2
;  1141          2173  2 ! Initialize the resource name.
;  1142          2174  2 !
;  1143          2175  2 RESNAM[0,0,32,0] = 'JBC$';
;  1144          2176  2 COPY_SYSID(.SYSID, RESNAM[4,0,0,0]);
;  1145          2177  2 RESNAM_DESC[0] = %ALLOCATION(RESNAM);
;  1146          2178  2 RESNAM_DESC[1] = RESNAM;
;  1147          2179  2
;  1148          2180  2
;  1149          2181  2 ! Enqueue for the doorbell lock of the remote system.
;  1150          2182  2 !
;  1151        P 2183  2 STATUS = $ENQ(
;  1152        P 2184  2     LKMODE=LCK$K_EXMODE,
;  1153        P 2185  2     LKSB=.LKSB,
;  1154        P 2186  2     RESNAM=RESNAM_DESC,
;  1155        P 2187  2     FLAGS=LCK$M_SYNCSTS OR LCK$M_NODLCKWT,
```

```
;  1156        P 2188  2      ASTADR=ENTER_REMOTE_REQUEST_AST,
;  1157          2189  2      ASTPRM=.LKSB);
;  1158          2190  2
;  1159          2191  2
;  1160          2192  2  ! Set flag is there is no doorbell lock defined for the remote job controller.
;  1161          2193  2  ! This indicates that either the remote node is not available (or not in the
;  1162          2194  2  ! cluster) or the remote job controller does not have the queue file open).
;  1163          2195  2  ! In either case the remote job controller may not respond for a long time or
;  1164          2196  2  ! possibly never (if an invalid node name were specified, for example).
;  1165          2197  2  !
;  1166          2198  2  ! Note that if SS$_SYNCH is set, the AST will not be delivered. Consequently,
;  1167          2199  2  ! lock dequeuing and memory deallocation must be performed here when the AST
;  1168          2200  2  ! routine is not executed.
;  1169          2201  2  !
;  1170          2202  2  FLAGS[FLAGS_V_NO_REMOTE_DOORBELL] = FALSE;
;  1171          2203  2  IF .STATUS EQL SS$_SYNCH
;  1172          2204  2  THEN
;  1173          2205  3      BEGIN
;  1174          2206  3      FLAGS[FLAGS_V_NO_REMOTE_DOORBELL] = TRUE;
;  1175          2207  3      $DEQ(LKID=.LKSB[4,0,32,0]);
;  1176          2208  3      DEALLOCATE_MEMORY(.LKSB);
;  1177          2209  2      END;
;  1178          2210  2
;  1179          2211  2
;  1180          2212  2  ! Check for service failure.
;  1181          2213  2  !
;  1182          2214  2  IF NOT .STATUS
;  1183          2215  2  THEN
;  1184          2216  2      SIGNAL(JBC$_COMREMJBC OR STS$K_ERROR, 0, .STATUS);
;  1185          2217  1  END;


                                                .EXTRN  SYS$DEQ

                                        001C 00000 ENTER_REMOTE_REQUEST:
                                                .WORD   Save R2,R3,R4                   ; 2131
                    54 00000000'    EF  9E 00002        MOVAB   FLAGS, R4
                    5E              14  C2 00009        SUBL2   #20, SP
00000000G           EF              00  FB 0000C        CALLS   #0, ALLOCATE_MEMORY     ; 2170
                    52              50  D0 00013        MOVL    R0, LKSB
                    6E  2443424A    8F  D0 00016        MOVL    #608387658, RESNAM      ; 2175
                    50        04    AC  D0 0001D        MOVL    SYSID, R0               ; 2176
              04    AE              60  D0 00021        MOVL    (R0), RESNAM+4
              08    AE        04    A0  B0 00025        MOVW    4(R0), RESNAM+8
              0C    AE              0A  D0 0002A        MOVL    #10, RESNAM_DESC        ; 2177
              10    AE              6E  9E 0002E        MOVAB   RESNAM, RESNAM_DESC+4   ; 2178
                                    7E  7C 00032        CLRQ    -(SP)                   ; 2189
                                    7E  D4 00034        CLRL    -(SP)
                                    52  DD 00036        PUSHL   LKSB
                          0000V     CF  9F 00038        PUSHAB  ENTER_REMOTE_REQUEST_AST
                                    7E  D4 0003C        CLRL    -(SP)
                            24      AE  9F 0003E        PUSHAB  RESNAM_DESC
                    7E      0208    8F  3C 00041        MOVZWL  #520, -(SP)
                                    52  DD 00046        PUSHL   LKSB
                                    05  DD 00048        PUSHL   #5
                                    7E  D4 0004A        CLRL    -(SP)
```

```
                    00000000G  00          0B  FB 0004C          CALLS   #11, SYS$ENQ                        ;
                               53          50  D0 00053          MOVL    R0, STATUS                          ;
                               64          10  8A 00056          BICB2   #16, FLAGS                          ; 2202
                    00000689   8F          53  D1 00059          CMPL    STATUS, #1673                       ; 2203
                                           1A  12 00060          BNEQ    1$                                  ;
                               64          10  88 00062          BISB2   #16, FLAGS                          ; 2206
                                           7E  7C 00065          CLRQ    -(SP)                               ; 2207
                                           7E  D4 00067          CLRL    -(SP)                               ;
                                       04  A2  DD 00069          PUSHL   4(LKSB)                             ;
                    00000000G  00          04  FB 0006C          CALLS   #4, SYS$DEQ                         ;
                                           52  DD 00073          PUSHL   LKSB                                ; 2208
                    00000000G  EF          01  FB 00075          CALLS   #1, DEALLOCATE_MEMORY               ;
                               11          53  E8 0007C 1$:      BLBS    STATUS, 2$                          ; 2214
                                           53  DD 0007F          PUSHL   STATUS                              ; 2216
                                           7E  D4 00081          CLRL    -(SP)                               ;
                                00048412   8F  DD 00083          PUSHL   #295954                             ;
                    00000000G  00          03  FB 00089          CALLS   #3, LIB$SIGNAL                      ;
                                               04 00090 2$:      RET                                         ; 2217

; Routine Size:  145 bytes,    Routine Base:  CODE + 06C2
```

```
;  1187          2218  1 ROUTINE ENTER_REMOTE_REQUEST_AST(LKSB): NOVALUE=
;  1188          2219  1
;  1189          2220  1 !++
;  1190          2221  1 !
;  1191          2222  1 ! FUNCTIONAL DESCRIPTION:
;  1192          2223  1 !       This routine is the completion AST routine for obtaining another job
;  1193          2224  1 !       controller's remote request lock.
;  1194          2225  1 !
;  1195          2226  1 ! INPUT PARAMETERS:
;  1196          2227  1 !       LKSB            - Pointer to LKSB allocated from dynamic memory.
;  1197          2228  1 !
;  1198          2229  1 ! IMPLICIT INPUTS:
;  1199          2230  1 !       NONE
;  1200          2231  1 !
;  1201          2232  1 ! OUTPUT PARAMETERS:
;  1202          2233  1 !       NONE
;  1203          2234  1 !
;  1204          2235  1 ! IMPLICIT OUTPUTS:
;  1205          2236  1 !       NONE
;  1206          2237  1 !
;  1207          2238  1 ! ROUTINE VALUE:
;  1208          2239  1 !       NONE
;  1209          2240  1 !
;  1210          2241  1 ! SIDE EFFECTS:
;  1211          2242  1 !       NONE
;  1212          2243  1 !
;  1213          2244  1 !--
;  1214          2245  1
;  1215          2246  2 BEGIN
;  1216          2247  2 MAP
;  1217          2248  2         LKSB:           REF BBLOCK;     ! Pointer to lock status block
;  1218          2249  2
;  1219          2250  2
;  1220          2251  2 ! Check status of the $ENQ.
;  1221          2252  2 !
;  1222          2253  2 IF NOT .LKSB[0,0,16,0]
;  1223          2254  2 THEN
;  1224          2255  2     SIGNAL(JBC$_COMREMJBC OR STS$K_ERROR, 0, .LKSB[0,0,16,0]);
;  1225          2256  2
;  1226          2257  2
;  1227          2258  2 ! Release the lock to enable the receiving job controller to recover it.
;  1228          2259  2 !
;  1229          2260  2 $DEQ(LKID=.LKSB[4,0,32,0]);
;  1230          2261  2
;  1231          2262  2
;  1232          2263  2 ! Deallocate the LKSB.
;  1233          2264  2 !
;  1234          2265  2 DEALLOCATE_MEMORY(.LKSB);
;  1235          2266  1 END;


                                0004 00000 ENTER_REMOTE_REQUEST_AST:
                                           .WORD   Save R2                                 ; 2218
                    52    04    AC  D0 00002         MOVL    LKSB, R2                     ; 2253
```

```
                        12          62  E8 00006          BLBS    (R2), 1$                                ;
                        7E          62  3C 00009          MOVZWL  (R2), -(SP)                             ; 2255
                                    7E  D4 0000C          CLRL    -(SP)                                   ;
                            00048412    8F  DD 000CE          PUSHL   #295954                             ;
                00000000G   00          03  FB 00014          CALLS   #3, LIB$SIGNAL                      ;
                                    7E  7C 0001B 1$:      CLRQ    -(SP)                                   ; 2260
                                    7E  D4 0001D          CLRL    -(SP)                                   ;
                                    04  A2  DD 0001F          PUSHL   4(R2)                               ;
                00000000G   00          04  FB 00022          CALLS   #4, SYS$DEQ                         ;
                                    52  DD 00029          PUSHL   R2                                      ; 2265
                00000000G   EF          01  FB 0002B          CALLS   #1, DEALLOCATE_MEMORY               ;
                                        04 00032          RET                                             ; 2266

; Routine Size:  51 bytes,    Routine Base:  CODE + 0753
```

```
;  1237          2267  1 GLOBAL ROUTINE QUEUE_MASTER_AST: NOVALUE=
;  1238          2268  1
;  1239          2269  1 !++
;  1240          2270  1 !
;  1241          2271  1 ! FUNCTIONAL DESCRIPTION:
;  1242          2272  1 !       This routine is the completion AST routine for the queue master lock.
;  1243          2273  1 !       It is entered when the queue master job controller fails and releases
;  1244          2274  1 !       the lock.
;  1245          2275  1 !
;  1246          2276  1 ! INPUT PARAMETERS:
;  1247          2277  1 !       Standard AST routine parameters (not used).
;  1248          2278  1 !
;  1249          2279  1 ! IMPLICIT INPUTS:
;  1250          2280  1 !       NONE
;  1251          2281  1 !
;  1252          2282  1 ! OUTPUT PARAMETERS:
;  1253          2283  1 !       NONE
;  1254          2284  1 !
;  1255          2285  1 ! IMPLICIT OUTPUTS:
;  1256          2286  1 !       NONE
;  1257          2287  1 !
;  1258          2288  1 ! ROUTINE VALUE:
;  1259          2289  1 !       NONE
;  1260          2290  1 !
;  1261          2291  1 ! SIDE EFFECTS:
;  1262          2292  1 !       NONE
;  1263          2293  1 !
;  1264          2294  1 !--
;  1265          2295  1
;  1266          2296  2 BEGIN
;  1267          2297  2
;  1268          2298  2 ! Ensure that at least one job controller holds a timer on the timed job
;  1269          2299  2 ! queue.
;  1270          2300  2 !
;  1271          2301  2 AFTER_AST();
;  1272          2302  1 END;


                                                0000 00000          .ENTRY  QUEUE_MASTER_AST, Save nothing          ; 2267
                      00000000G   EF          00   FB 00002         CALLS   #0, AFTER_AST                           ; 2301
                                                04 00009          RET                                             ; 2302

; Routine Size:  10 bytes,    Routine Base:  CODE + 0786
```

```
; 1274          2303  1 END
; 1275          2304  0 ELUDOM


                                                        .EXTRN  LIB$SIGNAL

;                                  PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
;  COMMON                         5024  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  OVR,NOPIC,ALIGN(2)
;  CODE                           1936  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)


;                                Library Statistics
;
;                                        -------- Symbols --------      Pages       Processing
;       File                             Total    Loaded   Percent      Mapped      Time
;
;  _$255$LUA28:[SYSLIB]LIB.L32;1          18619        53         0      1000          00:01.4


; Information:  2
; Warnings:     0
; Errors:       0


;                                       COMMAND QUALIFIERS

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

; Size:          1886 code + 5074 data bytes
; Run Time:        00:34.2
; Elapsed Time:    03:54.5
; Lines/CPU Min:    4040
; Lexemes/CPU-Min: 39299
; Memory Used:  380 pages
; Compilation Complete
```

BATCH
LIS

BROADCAST
LIS

BUFFERS
LIS

CONTROL
LIS

ASYNCHRON
LIS

CHECKPROT
LIS