

Figure 1 - Basic EZASOKET call paragraphs

```

* COMMENT LINES INDICATE MOVES THAT SHOULD BE DONE
EITHER
* BEFORE OR AFTER THE PARAGRAPH IS PERFORMED.
1 EZA-INITAPI.
2 MOVE 'INITAPI' TO EZA-FUNCTION.
3 MOVE +0 TO EZA-MAXSOC EZA-MAXSNO.
4 MOVE SPACES TO EZA-IDENT EZA-SUBTASK.
5 MOVE +0 TO EZA-ERRNO EZA-RETCODE.
6 CALL 'EZASOKET' USING EZA-FUNCTION
7 EZA-MAXSOC EZA-IDENT EZA-SUBTASK
8 EZA-MAXSNO
9 EZA-ERRNO EZA-RETCODE.
10 EZA-SOCKET.
11 MOVE 'SOCKET' TO EZA-FUNCTION.
12 MOVE +2 TO EZA-AF.
13 MOVE +1 TO EZA-SOCTYPE.
14 MOVE +0 TO EZA-PROTO.
15 MOVE +0 TO EZA-ERRNO EZA-RETCODE.
16 CALL 'EZASOKET' USING EZA-FUNCTION
17 EZA-AF EZA-SOCTYPE EZA-PROTO
18 EZA-ERRNO EZA-RETCODE.
19 * MOVE EZA-RETCODE TO EZA-S.
20 EZA-CONNECT.
21 * MOVE LOW-VALUES TO EZA-NAME.
22 * MOVE +2 TO EZA-NAME-FAMILY.
23 * MOVE WS-IPPORT TO EZA-NAME-PORT.
24 * MOVE WS-IPADDRESS TO EZA-NAME-IPADDRESS.
25 MOVE 'CONNECT' TO EZA-FUNCTION.
26 MOVE +0 TO EZA-ERRNO EZA-RETCODE.
27 CALL 'EZASOKET' USING EZA-FUNCTION
28 EZA-S EZA-NAME
29 EZA-ERRNO EZA-RETCODE.
30 EZA-SEND.
31 * MOVE WS-SEND-DATA TO BUFFER-OUT.
32 * MOVE LENGTH OF WS-SEND-DATA TO EZA-NBYTE.
33 * PERFORM EZA-EBCDIC-TO-ASCII.
34 MOVE 'SEND' TO EZA-FUNCTION.
35 MOVE +0 TO EZA-FLAGS.
36 MOVE +0 TO EZA-ERRNO EZA-RETCODE.
37 CALL 'EZASOKET' USING EZA-FUNCTION
38 EZA-S EZA-FLAGS EZA-NBYTE BUFFER-OUT
39 EZA-ERRNO EZA-RETCODE.
40 EZA-RECV.
41 MOVE 'RECV' TO EZA-FUNCTION.
42 MOVE +0 TO EZA-FLAGS.
43 MOVE SPACES TO BUFFER-IN.
44 MOVE LENGTH OF BUFFER-IN TO EZA-NBYTE.
45 MOVE +0 TO EZA-ERRNO EZA-RETCODE.
46 CALL 'EZASOKET' USING EZA-FUNCTION
47 EZA-S EZA-FLAGS EZA-NBYTE
48 BUFFER-IN
49 EZA-ERRNO EZA-RETCODE.
50 * MOVE EZA-RETCODE TO EZA-NBYTE.
51 * PERFORM EZA-ASCII-TO-EBCDIC.
52 EZA-EBCDIC-TO-ASCII.
53 CALL 'EZACIC04' USING BUFFER-OUT EZA-NBYTE.
54 EZA-ASCII-TO-EBCDIC.
55 CALL 'EZACIC05' USING BUFFER-IN EZA-NBYTE.
56 EZA-CLOSE.
57 MOVE 'CLOSE' TO EZA-FUNCTION.

```

```

58 MOVE +0 TO EZA-ERRNO EZA-RETCODE.
59 CALL 'EZASOKET' USING EZA-FUNCTION
60 EZA-S
61 EZA-ERRNO EZA-RETCODE.
62 EZA-TERMAPI.
63 MOVE 'TERMAPI' TO EZA-FUNCTION.
64 CALL 'EZASOKET' USING EZA-FUNCTION.
65 EZA-SELECT.
66 * MOVE LOW-VALUES TO EZA-TIMEOUT.
67 * MOVE 10 TO EZA-TIMEOUT-SECONDS.
68 * MOVE LOW-VALUES TO RSNDSK.
69 * MOVE LOW-VALUES TO WSNDSK.
70 * MOVE LOW-VALUES TO ESNDSK.
71 * COMPUTE EZA-MAXSOC-SELECT = EZA-S + 1.
72 * COMPUTE EZ6-SUB1 = EZA-S + 1.
73 * MOVE '1' TO EZ6-FLAG (EZ6-SUB1).
74 MOVE 'SELECT' TO EZA-FUNCTION.
75 MOVE LOW-VALUES TO RRETMSK WRETMSK ERETMSK.
76 MOVE +0 TO EZA-ERRNO EZA-RETCODE.
77 CALL 'EZASOKET' USING EZA-FUNCTION
78 EZA-MAXSOC-SELECT EZA-TIMEOUT
79 RSNDSK WSNDSK ESNDSK
80 RRETMSK WRETMSK ERETMSK
81 EZA-ERRNO EZA-RETCODE.
82 * MOVE EZ6-MASK TO RSNDSK (1 : LENGTH OF EZ6-MASK).
83 * PERFORM EZA-SELECT.
84 * IF EZA-RETCODE IS LESS THAN +0
85 * PERFORM ERROR-ROUTINE.
86 * IF EZA-RETCODE IS EQUAL TO +0
87 * PERFORM TIMEOUT-ERROR-ROUTINE.
88 * MOVE LOW-VALUES TO EZ6-MASK.
89 * MOVE LENGTH OF EZ6-MASK TO EZ6-SUB1.
90 * IF LENGTH OF ERETMSK IS LESS THAN EZ6-SUB1
91 * MOVE LENGTH OF ERETMSK TO EZ6-SUB1.
92 * MOVE RRETMSK TO EZ6-MASK (1 : EZ6-SUB1).
93 * PERFORM EZA-BITS-TO-CHARACTERS.
94 * COMPUTE EZ6-SUB1 = EZA-S + 1.
95 * IF EZ6-FLAG (EZ6-SUB1) IS NOT EQUAL TO '1'
96 * PERFORM ERROR-ROUTINE.
97 EZA-CHARACTERS-TO-BITS.
98 MOVE 'EZACIC06 CTB' TO CURRENT-FUNCTION.
99 MOVE LOW-VALUES TO EZ6-MASK.
100 MOVE LENGTH OF EZ6-FLAGS TO EZ6-MASK-LENGTH.
101 MOVE +0 TO EZA-RETCODE.
102 CALL 'EZACIC06' USINGEZ6-CTOB
103 EZ6-MASK EZ6-FLAGS EZ6-MASK-LENGTH
104 EZA-RETCODE.
105 EZA-BITS-TO-CHARACTERS.
106 MOVE ZEROES TO EZ6-FLAGS.
107 MOVE LENGTH OF EZ6-FLAGS TO EZ6-MASK-LENGTH.
108 MOVE +0 TO EZA-RETCODE.
109 CALL 'EZACIC06' USINGEZ6-BTOC
110 EZ6-MASK EZ6-FLAGS EZ6-MASK-LENGTH
111 EZA-RETCODE.
112 EZA-BIND.
113 * MOVE LOW-VALUES TO EZA-NAME.
114 * MOVE +2 TO EZA-NAME-FAMILY.
115 * MOVE WS-IPPORT TO EZA-NAME-PORT.
116 * MOVE WS-IPADDRESS TO EZA-NAME-IPADDRESS.
117 MOVE 'BIND' TO EZA-FUNCTION.
118 MOVE +0 TO EZA-ERRNO EZA-RETCODE.
119 CALL 'EZASOKET' USING EZA-FUNCTION

```

```

120         EZA-S EZA-NAME
121         EZA-ERRNO EZA-RETCODE.
122 EZA-LISTEN.
123         MOVE 'LISTEN' TO EZA-FUNCTION.
124         MOVE +0 TO EZA-BACKLOG.
125         MOVE +0 TO EZA-ERRNO EZA-RETCODE.
126         CALL 'EZASOKET' USING EZA-FUNCTION
127             EZA-S EZA-BACKLOG
128             EZA-ERRNO EZA-RETCODE.
129 EZA-ACCEPT.
130         MOVE 'ACCEPT' TO EZA-FUNCTION.
131         PERFORM CONSOLE-TRACE.
132         MOVE LOW-VALUES TO EZA-NAME.
133         MOVE +0 TO EZA-ERRNO EZA-RETCODE.
134         CALL 'EZASOKET' USING EZA-FUNCTION
135             EZA-S
136             EZA-NAME
137             EZA-ERRNO EZA-RETCODE.
138 EZA-PEEK.
139         MOVE 'RECV' TO EZA-FUNCTION.
140         MOVE +2 TO EZA-FLAGS.
141         MOVE LENGTH OF PEEK-BUFFER TO EZA-NBYTE.
142         MOVE +0 TO EZA-ERRNO EZA-RETCODE.
143         CALL 'EZASOKET' USING EZA-FUNCTION
144             EZA-S EZA-FLAGS EZA-NBYTE
145             PEEK-BUFFER
146             EZA-ERRNO EZA-RETCODE.
147 EZA-GETPEERNAME.
148         MOVE 'GETPEERNAME' TO EZA-FUNCTION.
149         MOVE +0 TO EZA-ERRNO EZA-RETCODE.
150         CALL 'EZASOKET' USING EZA-FUNCTION
151             EZA-S
152             EZA-NAME
153             EZA-ERRNO EZA-RETCODE.
154 EZA-RECVFROM.
155         MOVE 'RECVFROM' TO EZA-FUNCTION.
156         MOVE +0 TO EZA-FLAGS.
157         MOVE LENGTH OF BUFFER-IN TO EZA-NBYTE.
158         MOVE +0 TO EZA-ERRNO EZA-RETCODE.
159         CALL 'EZASOKET' USING EZA-FUNCTION
160             EZA-S EZA-FLAGS EZA-NBYTE
161             BUFFER-IN EZA-NAME
162             EZA-ERRNO EZA-RETCODE.
163 EZA-PEEKFROM.
164         MOVE 'RECVFROM' TO EZA-FUNCTION.
165         MOVE +2 TO EZA-FLAGS.
166         MOVE LENGTH OF PEEK-BUFFER TO EZA-NBYTE.
167         MOVE +0 TO EZA-ERRNO EZA-RETCODE.
168         CALL 'EZASOKET' USING EZA-FUNCTION
169             EZA-S EZA-FLAGS EZA-NBYTE
170             PEEK-BUFFER EZA-NAME
171             EZA-ERRNO EZA-RETCODE.
172 EZA-GETCLIENTID.
173         MOVE 'GETCLIENTID' TO EZA-FUNCTION.
174         MOVE LOW-VALUES TO EZA-CLIENT.
175         MOVE +0 TO EZA-ERRNO EZA-RETCODE.
176         CALL 'EZASOKET' USING EZA-FUNCTION
177             EZA-CLIENT
178             EZA-ERRNO EZA-RETCODE.
179 EZA-GIVESOCKET.
180         MOVE 'GIVESOCKET' TO EZA-FUNCTION.
181         MOVE SPACES TO EZA-CLIENT.

```

```

182         MOVE +2 TO EZA-CLIENT-DOMAIN.
183         MOVE +0 TO EZA-ERRNO EZA-RETCODE.
184         CALL 'EZASOKET' USING EZA-FUNCTION
185             EZA-S
186             EZA-CLIENT
187             EZA-ERRNO EZA-RETCODE.
188 EZA-TAKESOCKET.
189 *         MOVE START-CLIENTID TO EZA-CLIENT.
190 *         MOVE START-S TO EZA-S.
191 *         MOVE EZA-S TO EZA-SOCCRECV
192         MOVE 'TAKESOCKET' TO EZA-FUNCTION.
193         MOVE +0 TO EZA-ERRNO EZA-RETCODE.
194         CALL 'EZASOKET' USING EZA-FUNCTION
195             EZA-SOCCRECV
196             EZA-CLIENT
197             EZA-ERRNO EZA-RETCODE.
198 *         MOVE EZA-RETCODE TO EZA-S.

```

Figure 2 - Simple Client

```

1  MAINLINE SECTION.
2  ML-INITAPI.
3      PERFORM EZA-INITAPI.
4      IF EZA-RETCODE IS LESS THAN +0
5          PERFORM ERROR-ROUTINE.
6  ML-SOCKET.
7      PERFORM EZA-SOCKET.
8      IF EZA-RETCODE IS LESS THAN +0
9          PERFORM ERROR-ROUTINE.
10     MOVE EZA-RETCODE TO EZA-S.
11 ML-CONNECT.
12     MOVE LOW-VALUES TO EZA-NAME.
13     MOVE +2 TO EZA-NAME-FAMILY.
14     MOVE WS-IPPORT TO EZA-NAME-PORT.
15     MOVE WS-IPADDRESS TO EZA-NAME-IPADDRESS.
16     PERFORM EZA-CONNECT.
17     IF EZA-RETCODE IS LESS THAN +0
18         PERFORM ERROR-ROUTINE.
19 ML-SEND.
20     MOVE WS-SEND-DATA TO BUFFER-OUT.
21     MOVE LENGTH OF WS-SEND-DATA TO EZA-NBYTE.
22     PERFORM EZA-EBCDIC-TO-ASCII.
23     PERFORM EZA-SEND.
24     IF EZA-RETCODE IS LESS THAN +0
25         PERFORM ERROR-ROUTINE.
26 ML-RECV.
27     PERFORM EZA-RECV.
28     IF EZA-RETCODE IS LESS THAN +0
29         PERFORM ERROR-ROUTINE.
30     MOVE EZA-RETCODE TO EZA-NBYTE.
31     PERFORM EZA-ASCII-TO-EBCDIC.
32 ML-CLOSE.
33     PERFORM EZA-CLOSE.
34     IF EZA-RETCODE IS LESS THAN +0
35         PERFORM ERROR-ROUTINE.
36 ML-TERMAPI.
37     PERFORM EZA-TERMAPI.
38 ML-CICS-RETURN.
39     EXEC CICS RETURN END-EXEC
40 ML-EXIT.
41     GOBACK.

```

Figure 3 - Simple Server

```

1  MAINLINE SECTION.
2  ML-INITAPI.
3      PERFORM EZA-INITAPI.
4      IF EZA-RETCODE IS LESS THAN +0
5          PERFORM ERROR-ROUTINE.
6  ML-SOCKET.
7      PERFORM EZA-SOCKET.
8      IF EZA-RETCODE IS LESS THAN +0
9          PERFORM ERROR-ROUTINE.
10     MOVE EZA-RETCODE TO EZA-S.
11     MOVE EZA-RETCODE TO WS-SOCKET-ORIGINAL.
12 ML-BIND.
13     MOVE LOW-VALUES TO EZA-NAME.
14     MOVE +2 TO EZA-NAME-FAMILY.
15     MOVE WS-IPPORT TO EZA-NAME-PORT.
16     MOVE WS-IPADDRESS TO EZA-NAME-IPADDRESS.
17     PERFORM EZA-BIND.
18     IF EZA-RETCODE IS LESS THAN +0
19         PERFORM ERROR-ROUTINE.
20 ML-LISTEN.
21     PERFORM EZA-LISTEN.
22     IF EZA-RETCODE IS LESS THAN +0
23         PERFORM ERROR-ROUTINE.
24 ML-ACCEPT.
25     MOVE WS-SOCKET-ORIGINAL TO EZA-S.
26     PERFORM EZA-ACCEPT.
27     IF EZA-RETCODE IS LESS THAN +0
28         PERFORM ERROR-ROUTINE.
29     MOVE EZA-RETCODE TO EZA-S.
30     MOVE EZA-RETCODE TO WS-SOCKET-ACCEPT.
31 ML-RECV.
32     PERFORM EZA-RECV.
33     IF EZA-RETCODE IS LESS THAN +0
34         PERFORM ERROR-ROUTINE.
35     MOVE EZA-RETCODE TO EZA-NBYTE.
36     PERFORM EZA-ASCII-TO-EBCDIC.
37 ML-SEND.
38     MOVE WS-SEND-DATA TO BUFFER-OUT.
39     MOVE LENGTH OF WS-SEND-DATA TO EZA-NBYTE.
40     PERFORM EZA-EBCDIC-TO-ASCII.
41     PERFORM EZA-SEND.
42     IF EZA-RETCODE IS LESS THAN +0
43         PERFORM ERROR-ROUTINE.
44 ML-CLOSE.
45     PERFORM EZA-CLOSE.
46     IF EZA-RETCODE IS LESS THAN +0
47         PERFORM ERROR-ROUTINE.
48 ML-DOAGAIN.
49     GO TO ML-ACCEPT.
50 ML-CLOSE-ORIGINAL.
51     PERFORM EZA-CLOSE.
52     IF EZA-RETCODE IS LESS THAN +0
53         PERFORM ERROR-ROUTINE.
54 ML-TERMAPI.
55     PERFORM EZA-TERMAPI.
56 ML-CICS-RETURN.
57     EXEC CICS RETURN END-EXEC
58 ML-EXIT.
59     GOBACK.

```

Figure 4 - Basic SSELECT logic

```

1  ML-SELECT.
2      MOVE LOW-VALUES TO EZA-TIMEOUT.
3      MOVE 10 TO EZA-TIMEOUT-SECONDS.
4      MOVE LOW-VALUES TO RSNDMSK.
5      MOVE LOW-VALUES TO WSNDMSK.
6      MOVE LOW-VALUES TO ESNDMSK.
7      COMPUTE EZA-MAXSOC-SELECT = EZA-S + 1.
8      COMPUTE EZ6-SUB1 = EZA-S + 1.
9      MOVE '1' TO EZ6-FLAG (EZ6-SUB1).
10     PERFORM EZA-CHARACTERS-TO-BITS.
11     MOVE EZ6-MASK TO RSNDMSK (1 : LENGTH OF EZ6-MASK).
12     PERFORM EZA-SELECT.
13     IF EZA-RETCODE IS LESS THAN +0
14         PERFORM ERROR-ROUTINE.
15     IF EZA-RETCODE IS EQUAL TO +0
16         PERFORM TIMEOUT-ERROR-ROUTINE.
17     MOVE LOW-VALUES TO EZ6-MASK.
18     MOVE LENGTH OF EZ6-MASK TO EZ6-SUB1.
19     IF LENGTH OF ERETMSK IS LESS THAN EZ6-SUB1
20         MOVE LENGTH OF ERETMSK TO EZ6-SUB1.
21     MOVE RRETMSK TO EZ6-MASK (1 : EZ6-SUB1).
22     PERFORM EZA-BITS-TO-CHARACTERS.
23     COMPUTE EZ6-SUB1 = EZA-S + 1.
24     IF EZ6-FLAG (EZ6-SUB1) IS NOT EQUAL TO '1'
25         PERFORM ERROR-ROUTINE.

```

Figure 6 - Child Program

```

1  MAINLINE SECTION.
2  ML-INITAPI.
3      PERFORM EZA-INITAPI.
4      IF EZA-RETCODE IS LESS THAN +0
5          PERFORM ERROR-ROUTINE.
6  ML-TAKESOCKET.
7      EXEC CICS RETRIEVE
8          INTO (START-CHILD-PARM)
9              LENGTH (LENGTH OF START-CHILD-PARM)
10             NOHANDLE
11     END-EXEC.
12     MOVE START-CLIENTID TO EZA-CLIENT.
13     MOVE START-S TO EZA-S.
14     MOVE EZA-S TO EZA-SOCRECV.
15     PERFORM EZA-TAKESOCKET.
16     IF EZA-RETCODE IS LESS THAN +0
17         PERFORM ERROR-ROUTINE.
18     MOVE EZA-RETCODE TO EZA-S.
19 ML-RECV.
20     PERFORM EZA-RECV.
21     IF EZA-RETCODE IS LESS THAN +0
22         PERFORM ERROR-ROUTINE.
23     MOVE EZA-RETCODE TO EZA-NBYTE.
24     PERFORM EZA-ASCII-TO-EBCDIC.
25 ML-SEND.
26     MOVE WS-SEND-DATA TO BUFFER-OUT.
27     MOVE LENGTH OF WS-SEND-DATA TO EZA-NBYTE.
28     PERFORM EZA-EBCDIC-TO-ASCII.
29     PERFORM EZA-SEND.
30     IF EZA-RETCODE IS LESS THAN +0
31         PERFORM ERROR-ROUTINE.
32 ML-CLOSE.

```

```

33     PERFORM EZA-CLOSE.
34     IF EZA-RETCODE IS LESS THAN +0
35         PERFORM ERROR-ROUTINE.
36 ML-TERMAPI.
37     PERFORM EZA-TERMAPI.
38 ML-CICS-RETURN.
39     EXEC CICS RETURN END-EXEC
40 ML-EXIT.
41     GOBACK.

```

Figure 7 - Complex Server

```

1     01 SOCKET-FLAGS                VALUE SPACES.
2         05 SOCKET-STATUS          PIC X(01) OCCURS 64 TIMES.
3     * VALUES FOR SOCKET-STATUS CAN BE:
4     *     ' ' SOCKET IS UNUSED OR HAS BEEN TAKEN
5     *     'L' SOCKET IS IN LISTEN MODE
6     *     'P' SOCKET IS WAITING FOR PEEK DATA
7     *     'G' SOCKET IS BEING GIVEN
8     01 SOCKET-MASKS                VALUE SPACES.
9         05 SM-RECVS.
10            10 SM-RECV             PIC X(01) OCCURS 64 TIMES.
11            05 SM-WRITES.
12            10 SM-WRITE           PIC X(01) OCCURS 64 TIMES.
13            05 SM-EXCEPTIONS.
14            10 SM-EXCEPTION       PIC X(01) OCCURS 64 TIMES.
15     01 SOCKET-MASKS-LIMIT         PIC S9(04) COMP VALUE +64.
16     01 SOCKET-MASKS-USED         PIC S9(04) COMP VALUE +0.
17     01 START-CHILD-PARM.
18     * THIS AREA IS DOCUMENTED IN "LISTENER OUTPUT FORMAT",
19     * FIGURE 15 IN CHAPTER 14, EXTERNAL DATA STRUCTURES, OF
20     * "TCP/IP FOR VSE/ESA - IBM PROGRAM SETUP AND
21     * SUPPLEMENTARY INFORMATION - SC33-6601-5"
22         05 START-S                PIC 9(08) COMP.
23         05 START-CLIENTID.
24             15 START-C-DOMAIN     PIC 9(08) COMP.
25             15 START-C-NAME       PIC X(08).
26             15 START-C-TASK       PIC X(08).
27             15 FILLER             PIC X(20).
28         05 START-USER-DATA        PIC X(35).
29         05 FILLER                 PIC X(01).
30         05 START-NAME.
31             15 START-FAMILY       PIC 9(04) COMP.
32             15 START-PORT         PIC 9(04) COMP.
33             15 START-ADDRESS      PIC 9(08) COMP.
34             15 START-ZERO         PIC X(08).
35     01 CLIENT-DATA.
36     * THIS AREA IS DOCUMENTED IN "LISTENER INPUT FORMAT",
37     * IN CHAPTER 15, EXTERNAL DATA STRUCTURES, OF
38     * "TCP/IP FOR VSE/ESA - IBM PROGRAM SETUP AND
39     * SUPPLEMENTARY INFORMATION - SC33-6601-5"
40         05 CLIENT-TRANID          PIC X(04).
41         05 CLIENT-USER-DATA       PIC X(35).
42         05 CLIENT-START-METHOD  PIC X(02).
43         05 CLIENT-INTERVAL-TIME   PIC X(06).
44 MAINLINE SECTION.
45 ML-START.
46     PERFORM EZACIC06-SETUP.
47 ML-INITAPI.
48     MOVE SOCKET-MASKS-LIMIT TO EZA-MAXSNO.
49     PERFORM EZA-INITAPI.
50     IF EZA-RETCODE IS LESS THAN +0

```

```

52     PERFORM ERROR-ROUTINE.
53 ML-GETCLIENTID.
54     PERFORM EZA-GETCLIENTID.
55     IF EZA-RETCODE IS LESS THAN +0
56         PERFORM ERROR-ROUTINE.
57     MOVE EZA-CLIENT TO WS-OURCLIENT.
58 ML-SOCKET.
59     PERFORM EZA-SOCKET.
60     IF EZA-RETCODE IS LESS THAN +0
61         PERFORM ERROR-ROUTINE.
62     MOVE EZA-RETCODE TO EZA-S.
63     MOVE EZA-RETCODE TO WS-SOCKET-ORIGINAL.
64 ML-BIND.
65     MOVE LOW-VALUES TO EZA-NAME.
66     MOVE +2 TO EZA-NAME-FAMILY.
67     MOVE WS-IPPORT TO EZA-NAME-PORT.
68     MOVE WS-IPADDRESS TO EZA-NAME-IPADDRESS.
69     PERFORM EZA-BIND.
70     IF EZA-RETCODE IS LESS THAN +0
71         PERFORM ERROR-ROUTINE.
72 ML-LISTEN.
73     PERFORM EZA-LISTEN.
74     IF EZA-RETCODE IS LESS THAN +0
75         PERFORM ERROR-ROUTINE.
76     COMPUTE WS-SUB1 = EZA-S + 1.
77     MOVE 'L' TO SOCKET-STATUS (WS-SUB1).
78     PERFORM WAIT-FOR-DATA.
79     PERFORM CLOSE-ALL.
80 ML-TERMAPI.
81     PERFORM EZA-TERMAPI.
82     EXEC CICS RETURN END-EXEC.
83 ML-EXIT.
84     GOBACK.
85 WAIT-FOR-DATA SECTION.
86 WFD-START.
87     ADD +1 TO WS-COUNT.
88 WFD-SELECT-SET.
89     MOVE ZEROES TO SOCKET-MASKS.
90     MOVE -1 TO WS-CUR-SOCKET.
91 WFD-SELECT-SET-LOOP.
92     ADD +1 TO WS-CUR-SOCKET.
93     COMPUTE WS-SUB1 = WS-CUR-SOCKET + 1.
94     IF WS-SUB1 IS GREATER THAN SOCKET-MASKS-LIMIT
95         GO TO WFD-SELECT-SET-END.
96     IF SOCKET-STATUS (WS-SUB1) IS EQUAL TO ' '
97         GO TO WFD-SELECT-SET-LOOP.
98     MOVE WS-SUB1 TO SOCKET-MASKS-USED.
99     IF SOCKET-STATUS (WS-SUB1) IS EQUAL TO 'L'
100         MOVE '1' TO SM-RECV (WS-SUB1).
101     IF SOCKET-STATUS (WS-SUB1) IS EQUAL TO 'P'
102         MOVE '1' TO SM-RECV (WS-SUB1).
103     IF SOCKET-STATUS (WS-SUB1) IS EQUAL TO 'G'
104         MOVE '1' TO SM-EXCEPTION (WS-SUB1).
105     GO TO WFD-SELECT-SET-LOOP.
106 WFD-SELECT-SET-END.
107 WFD-SELECT.
108     MOVE LOW-VALUES TO EZA-TIMEOUT.
109     MOVE 30 TO EZA-TIMEOUT-SECONDS.
110     MOVE SOCKET-MASKS-USED TO EZA-MAXSOC-SELECT.
111
112     MOVE SM-RECVS TO EZ6-FLAGS.
113     PERFORM EZA-CHARACTERS-TO-BITS.

```

```

114 MOVE LOW-VALUES TO RSNDMSK.
115 MOVE EZ6-MASK TO RSNDMSK (1 : LENGTH OF EZ6-MASK).
116
117 MOVE SM-WRITES TO EZ6-FLAGS.
118 PERFORM EZA-CHARACTERS-TO-BITS.
119 MOVE LOW-VALUES TO WSNDMSK.
120 MOVE EZ6-MASK TO WSNDMSK (1 : LENGTH OF EZ6-MASK).
121 MOVE SM-EXCEPTIONS TO EZ6-FLAGS.
122 PERFORM EZA-CHARACTERS-TO-BITS.
123 MOVE LOW-VALUES TO ESNDMSK.
124 MOVE EZ6-MASK TO ESNDMSK (1 : LENGTH OF EZ6-MASK).
125 PERFORM EZA-SELECT.
126 IF EZA-RETCODE IS LESS THAN +0
127     PERFORM ERROR-ROUTINE.
128 IF EZA-RETCODE IS EQUAL TO +0
129     GO TO WFD-SELECT.
130 MOVE RRETMSK TO EZ6-MASK.
131 PERFORM EZA-BITS-TO-CHARACTERS.
132 MOVE EZ6-FLAGS TO SM-RECVS.
133 MOVE WRETMSK TO EZ6-MASK.
134 PERFORM EZA-BITS-TO-CHARACTERS.
135 MOVE EZ6-FLAGS TO SM-WRITES.
136 MOVE ERETMSK TO EZ6-MASK.
137 PERFORM EZA-BITS-TO-CHARACTERS.
138 MOVE EZ6-FLAGS TO SM-EXCEPTIONS.
139 WFD-SELECT-INSPECT.
140 MOVE -1 TO WS-CUR-SOCKET.
141 WFD-SELECT-INSPECT-LOOP.
142 ADD +1 TO WS-CUR-SOCKET.
143 COMPUTE WS-SUB1 = WS-CUR-SOCKET + 1.
144 IF WS-SUB1 IS GREATER THAN SOCKET-MASKS-LIMIT
145     GO TO WFD-SELECT-INSPECT-END.
146 IF SOCKET-STATUS (WS-SUB1) IS EQUAL TO ' '
147     GO TO WFD-SELECT-INSPECT-LOOP.
148 IF SM-RECV (WS-SUB1) IS EQUAL TO '1'
149     AND SOCKET-STATUS (WS-SUB1) IS EQUAL TO 'L'
150     PERFORM NEW-CONNECTION.
151 IF SM-RECV (WS-SUB1) IS EQUAL TO '1'
152     AND SOCKET-STATUS (WS-SUB1) IS EQUAL TO 'P'
153     PERFORM START-REQUESTED.
154 IF SM-EXCEPTION (WS-SUB1) IS EQUAL TO '1'
155     AND SOCKET-STATUS (WS-SUB1) IS EQUAL TO 'G'
156     PERFORM CLOSE-GIVEN.
157 GO TO WFD-SELECT-INSPECT-LOOP.
158 WFD-SELECT-INSPECT-END.
159 EXEC CICS INQUIRE SYSTEM
160     CICSSTATUS(WS-CVDA)
161 END-EXEC.
162 IF WS-CVDA IS NOT EQUAL TO DFHVALUE(ACTIVE)
163     AND WS-CVDA IS NOT EQUAL TO DFHVALUE(STARTUP)
164     MOVE 'Y' TO WS-SHUTDOWN.
165 IF SHUTDOWN-PROGRAM
166     GO TO WFD-EXIT.
167 GO TO WFD-START.
168 WFD-EXIT.
169 EXIT.
170 NEW-CONNECTION SECTION.
171 NC-ACCEPT.
172 MOVE WS-CUR-SOCKET TO EZA-S.
173 PERFORM EZA-ACCEPT.
174 IF EZA-RETCODE IS EQUAL TO -1
175 * OTHER END DROPPED BEFORE WE CONNECTED

176 AND EZA-ERRNO IS EQUAL TO +30358
177 GO TO NC-EXIT.
178 IF EZA-RETCODE IS LESS THAN +0
179     PERFORM ERROR-ROUTINE.
180 COMPUTE WS-SUB1 = EZA-RETCODE + 1.
181 MOVE 'P' TO SOCKET-STATUS (WS-SUB1).
182 NC-EXIT.
183 EXIT.
184 START-REQUESTED SECTION.
185 SR-RECVFROM.
186 MOVE WS-CUR-SOCKET TO EZA-S.
187 PERFORM EZA-RECVFROM.
188 IF EZA-RETCODE IS LESS THAN +0
189     PERFORM ERROR-ROUTINE.
190 MOVE EZA-RETCODE TO EZA-NBYTE.
191 PERFORM EZA-ASCII-TO-EBCDIC.
192 MOVE EZA-NBYTE TO WS-TEXT-LENGTH
193 MOVE EZA-NAME TO WS-CLIENT-NAME.
194 MOVE SPACES TO CLIENT-DATA.
195 UNSTRING BUFFER-IN (1:EZA-NBYTE) DELIMITED BY ',' INTO
196     CLIENT-TRANID
197     CLIENT-USER-DATA
198     CLIENT-START-METHOD
199     CLIENT-INTERVAL-TIME
200     ON OVERFLOW
201     GO TO SR-SECURITY.
202 SR-MAYBE-SHUTDOWN.
203 IF CLIENT-TRANID = 'DOWN'
204     OR CLIENT-TRANID = 'STOP'
205     MOVE 'Y' TO WS-SHUTDOWN
206     GO TO SR-EXIT.
207 SR-GIVESOCKET.
208 PERFORM EZA-GIVESOCKET.
209 IF EZA-RETCODE IS LESS THAN +0
210     PERFORM ERROR-ROUTINE.
211 MOVE WS-CUR-SOCKET TO WS-SOCKET-GIVE.
212 PERFORM SPAWN-CHILD.
213 SR-EXIT.
214 EXIT.
215 SPAWN-CHILD SECTION.
216 SC-START.
217 MOVE SPACES TO START-CHILD-PARM.
218 MOVE WS-SOCKET-ACCEPT TO START-S.
219 MOVE WS-OURCLIENT TO START-CLIENTID.
220 MOVE CLIENT-USER-DATA TO START-USER-DATA.
221 MOVE EZA-NAME TO START-NAME.
222 EXEC CICS START
223     TRANSID(CLIENT-TRANID)
224     FROM(START-CHILD-PARM)
225     LENGTH(LENGTH OF START-CHILD-PARM)
226     NOHANDLE
227 END-EXEC
228 COMPUTE WS-SUB1 = WS-SOCKET-GIVE + 1.
229 MOVE 'G' TO SOCKET-STATUS (WS-SUB1).
230 IF EIBRESP IS NOT EQUAL TO DFHRESP(NORMAL)
231     MOVE MSG-INVALID-REQUEST TO BUFFER-OUT
232     PERFORM INVALID-INFORMATION.
233 SC-EXIT.
234 EXIT.
235 CLOSE-GIVEN SECTION.
236 CG-START.
237 MOVE WS-CUR-SOCKET TO EZA-S.

```

```

238     PERFORM EZA-CLOSE.
239     IF EZA-RETCODE IS LESS THAN +0
240         PERFORM ERROR-ROUTINE.
241     COMPUTE WS-SUB1 = WS-CUR-SOCKET + 1.
242     MOVE ' ' TO SOCKET-STATUS (WS-SUB1).
243 CG-EXIT.
244     EXIT.
245 CLOSE-ALL SECTION.
246 CA-START.
247     MOVE -1 TO WS-CUR-SOCKET.
248 CA-SHUTDOWN-LOOP.
249     ADD +1 TO WS-CUR-SOCKET.
250     IF WS-CUR-SOCKET IS GREATER THAN SOCKET-MASKS-LIMIT
251         GO TO CA-EXIT.
252     IF SOCKET-STATUS (WS-CUR-SOCKET) IS EQUAL TO SPACE
253         GO TO CA-SHUTDOWN-LOOP.
254 CA-CLOSE.
255     MOVE WS-CUR-SOCKET TO EZA-S
256     PERFORM EZA-CLOSE
257     IF EZA-RETCODE IS LESS THAN +0
258         PERFORM ERROR-ROUTINE.
259     COMPUTE WS-SUB1 = WS-CUR-SOCKET + 1.
260     MOVE ' ' TO SOCKET-STATUS (WS-SUB1).
261     GO TO CA-SHUTDOWN-LOOP.
262 CA-EXIT.
263     EXIT.
264 INVALID-INFORMATION SECTION.
265 II-START.
266     MOVE LENGTH OF MSG-SCTY-ERROR TO EZA-NBYTE.
267     PERFORM EZA-EBCDIC-TO-ASCII.
268     PERFORM EZA-SEND.
269     IF HAVE-TERMINAL
270         EXEC CICS SEND TEXT
271             FROM(MSG-SCTY-ERROR)
272             LENGTH(LENGTH OF MSG-SCTY-ERROR)
273             ERASE
274             END-EXEC.
275     EXEC CICS WRITE OPERATOR
276         TEXT(MSG-SCTY-ERROR)
277         TEXTLENGTH(LENGTH OF MSG-SCTY-ERROR)
278     END-EXEC.
279     PERFORM EZA-CLOSE.
280     IF EZA-RETCODE IS LESS THAN +0
281         PERFORM ERROR-ROUTINE.
282     COMPUTE WS-SUB1 = WS-CUR-SOCKET + 1.
283     MOVE ' ' TO SOCKET-STATUS (WS-SUB1).
284 II-EXIT.
285     EXIT.
286 EZACIC06-SETUP SECTION.
287 EZ6S-START.
288     MOVE LENGTH OF EZ6-FLAGS TO EZ6-MAXSOC.
289     SUBTRACT +1 FROM EZ6-MAXSOC.
290     MOVE LENGTH OF EZ6-MASK TO EZ6-MASK-LENGTH.
291     COMPUTE EZ6-SUB1 = (EZ6-MASK-LENGTH * 8) - 1.
292     IF EZ6-SUB1 IS LESS THAN EZ6-MAXSOC
293         MOVE EZ6-SUB1 TO EZ6-MAXSOC.
294 EZ6S-EXIT.
295     EXIT.

```