

EEEEEEEEEEEEEEEE	DDDDDDDDDDDD		FFFFFFFFFFFFFF
EEEEEEEEEEEEEEEE	DDDDDDDDDDDD		FFFFFFFFFFFFFF
EEEEEEEEEEEEEEEE	DDDDDDDDDDDD		FFFFFFFFFFFFFF
EEE	DDD	DDD	FFF
EEE	DDD	DDD	FFF
EEE	DDD	DDD	FFF
EEE	DDD	DDD	FFF
EEE	DDD	DDD	FFF
EEE	DDD	DDD	FFF
EEEEEEEEEEEE	DDD	DDD	FFFFFFFFFFFF
EEEEEEEEEEEE	DDD	DDD	FFFFFFFFFFFF
EEEEEEEEEEEE	DDD	DDD	FFFFFFFFFFFF
EEE	DDD	DDD	FFF
EEE	DDD	DDD	FFF
EEE	DDD	DDD	FFF
EEE	DDD	DDD	FFF
EEE	DDD	DDD	FFF
EEEEEEEEEEEE	DDDDDDDDDDDD		FFF
EEEEEEEEEEEE	DDDDDDDDDDDD		FFF
EEEEEEEEEEEE	DDDDDDDDDDDD		FFF

_S
Va
--
00
00
00
00
00
00
00
00
00
00
7F
7F
7F
7F
7F
7F
7F
7F


```
0137 PROCEDURE CHECK_QUOTES (VAR DESC: DESCRIPTOR);
0138
0139 VAR
0140     SCAN_INDEX : INTEGER;
0141     INDEX      : INTEGER;
0142     QUOTE_FOUND : BOOLEAN;
0143     APOST_FOUND : BOOLEAN;
0144     QCHAR      : CHAR;
0145     QUOTES     : CHAR;
0146
0147 BEGIN
0148
0149     QUOTES := CHR (34);
0150
0151     { +
0152     Non-null strings are the most interesting.
0153     - }
0154     IF DESC.DSC$W_LENGTH <> 0 THEN
0155     BEGIN
0156
0157         { +
0158         Don't disturb a string that already is enclosed in (')s or (")s.
0159         - }
0160         IF NOT (
0161             (
0162                 (DESC.DSC$A_POINTER^[1] = APOSTROPHE)
0163                 AND
0164                 (DESC.DSC$A_POINTER^[DESC.DSC$W_LENGTH] = APOSTROPHE)
0165             )
0166             OR
0167             (
0168                 (DESC.DSC$A_POINTER^[1] = QUOTES)
0169                 AND
0170                 (DESC.DSC$A_POINTER^[DESC.DSC$W_LENGTH] = QUOTES)
0171             )
0172         ) THEN
0173
0174     BEGIN
0175
0176         { +
0177         First, see if there are any nasty embedded quotes or apostrophes.
0178         - }
0179         QUOTE_FOUND := FALSE;
0180         APOST_FOUND := FALSE;
0181
0182         FOR INDEX := 1 TO DESC.DSC$W_LENGTH DO
0183         BEGIN
0184
0185             IF DESC.DSC$A_POINTER^[INDEX] = QUOTES THEN
0186
0187                 QUOTE_FOUND := TRUE;
0188
0189             IF DESC.DSC$A_POINTER^[INDEX] = APOSTROPHE THEN
0190
0191                 APOST_FOUND := TRUE;
0192
0193
```

```
0194 END;
0195 { +
0196 Just bracket the string with quotes if no quotes are found,
0197 or if just bracket it with apostrophes if no apostrophes are
0198 found.
0199 - )
0200 IF QUOTE_FOUND THEN
0201 BEGIN
0202     IF APOST_FOUND THEN
0203     BEGIN
0204         SCAN_INDEX := DESC.DSC$W_LENGTH;
0205         REPEAT
0206             IF DESC.DSC$A_POINTER^[SCAN_INDEX] = QUOTES THEN
0207             BEGIN
0208                 FOR INDEX := DESC.DSC$W_LENGTH DOWNTO SCAN_INDEX DO
0209                     DESC.DSC$A_POINTER^[INDEX+1]
0210                     := DESC.DSC$A_POINTER^[INDEX];
0211                 IF DESC.DSC$W_LENGTH < 254 THEN
0212                     DESC.DSC$W_LENGTH := DESC.DSC$W_LENGTH + 1;
0213             END;
0214             SCAN_INDEX := SCAN_INDEX - 1;
0215         UNTIL SCAN_INDEX < 1;
0216         QCHAR := QUOTES;
0217     END { IF TRUE APOST_FOUND (AND QUOTE_FOUND) -YUK }
0218     ELSE
0219     BEGIN
0220         QCHAR := APOSTROPHE;
0221     END; { IF FALSE APOST_FOUND }
0222 END { IF TRUE QUOTE_FOUND }
0223 ELSE
0224 BEGIN
```

0194
0195
0196
0197
0198
0199
0200
0201
0202
0203
0204
0205
0206
0207
0208
0209
0210
0211
0212
0213
0214
0215
0216
0217
0218
0219
0220
0221
0222
0223
0224
0225
0226
0227
0228
0229
0230
0231
0232
0233
0234
0235
0236
0237
0238
0239
0240
0241
0242
0243
0244
0245
0246
0247
0248
0249
0250

ED
VC
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
00


```
0251     QCHAR           := QUOTES;
0252
0253     END;           { IF FALSE QUOTE_FOUND }
0254
0255     { +
0256     Shift the string down one char.
0257     - }
0258     FOR INDEX      := DESC.DSC$W_LENGTH DOWNT0 1 DO
0259
0260         DESC.DSC$A_POINTER^[INDEX+1] := DESC.DSC$A_POINTER^[INDEX];
0261
0262     { +
0263     Update the length and add the quotes.
0264     - }
0265     DESC.DSC$W_LENGTH := DESC.DSC$W_LENGTH + 2;
0266     DESC.DSC$A_POINTER^[1] := QCHAR;
0267     DESC.DSC$A_POINTER^[DESC.DSC$W_LENGTH] := QCHAR;
0268
0269     END;           { IF NOT (ALREADY QUOTED) }
0270
0271     END           { IF TRUE DESC.DSC$W_LENGTH <> 0 }
0272
0273     ELSE
0274
0275     BEGIN
0276
0277         STR$TRIM (DESC,EMPTY_STRING);
0278
0279     END;           { IF FALSE DESC.DSC$W_LENGTH <> 0 }
0280
0281     END;           { CHECK_QUOTES }
```



```
0328 PROCEDURE SHOW_PRIMARY;
0329
0330 VAR
0331     TEMP_PRI      : PRIMARY_TYPE;
0332
0333 BEGIN
0334     WITH DEF_CURRENT^ DO
0335     BEGIN
0336         { +
0337         Fetch the primary we're showing.
0338         - }
0339         TEMP_PRI      := PRIMARY;
0340
0341         { +
0342         Output it, using the correct width.
0343         - }
0344         WRITE (FDL_DEST,TEMP_PRI:PRIMARY_WIDTH[TEMP_PRI]);
0345
0346         { +
0347         If it's one with a PRINUM, put that out too.
0348         - }
0349         IF TEMP_PRI IN [ ANALYSIS_OF_AREA, ANALYSIS_OF_KEY, AREA, KEY ] THEN
0350             WRITE (FDL_DEST,' ',PRINUM:NUM_LEN(PRINUM));
0351
0352         { +
0353         If it's one with a string value, put that out too.
0354         - }
0355         IF TEMP_PRI IN [ IDENT, TITLE ] THEN
0356             BEGIN
0357                 CHECK_QUOTES (STRING);
0358                 IF (STRING.DSC$W_LENGTH > 0) THEN
0359                     WRITE (FDL_DEST,' ',STRING.DSC$A_POINTER^:
0360                         STRING.DSC$W_LENGTH);
0361             END;
0362
0363     END;
0364
0365 END;      { WITH DEF_CURRENT^ }
0366
0367 END;      { SHOW_PRIMARY }
```



```
0548 PROCEDURE SHOW_QUALIFIER;  
0549  
0550 BEGIN  
0551  
0552   { +  
0553   Output the qualifiers.  
0554   - }  
0555   CASE DEF_CURRENT^.QUALIFIER OF  
0556  
0557     FDLSC_ANYPOS :      WRITE (FDL_DEST,'any_cylinder');  
0558     FDLSC_CLUSPOS :    WRITE (FDL_DEST,'cluster');  
0559     FDLSC_CYLPOS :     WRITE (FDL_DEST,'cylinder');  
0560     FDLSC_FIDPOS :     WRITE (FDL_DEST,'file_ID');  
0561     FDLSC_FNMPOS :     WRITE (FDL_DEST,'file_name');  
0562     FDLSC_LOGPOS :     WRITE (FDL_DEST,'logical');  
0563     FDLSC_VIRPOS :     WRITE (FDL_DEST,'virtual');  
0564     FDLSC_NOPOS ,  
0565     FDLSC_NONE :      WRITE (FDL_DEST,'none');  
0566     FDLSC_IDX :       WRITE (FDL_DEST,'indexed');  
0567     FDLSC_REL :       WRITE (FDL_DEST,'relative');  
0568     FDLSC_SEQ :       WRITE (FDL_DEST,'sequential');  
0569     FDLSC_IF_IN :     WRITE (FDL_DEST,'if_in_recovery_unit');  
0570     FDLSC_NECESSARY : WRITE (FDL_DEST,'necessary_to_write');  
0571     FDLSC_NEVER :     WRITE (FDL_DEST,'never_RU_journal');  
0572     FDLSC_CR :        WRITE (FDL_DEST,'carriage_return');  
0573     FDLSC_FTN :       WRITE (FDL_DEST,'FORTRAN');  
0574     FDLSC_PRINT :     WRITE (FDL_DEST,'print');  
0575     FDLSC_FIX :       WRITE (FDL_DEST,'fixed');  
0576     FDLSC_STM :       WRITE (FDL_DEST,'stream');  
0577     FDLSC_STMCR :     WRITE (FDL_DEST,'stream_CR');  
0578     FDLSC_STMLF :     WRITE (FDL_DEST,'stream_LF');  
0579     FDLSC_UDF :       WRITE (FDL_DEST,'undefined');  
0580     FDLSC_VAR :       WRITE (FDL_DEST,'variable');  
0581     FDLSC_VFC :       WRITE (FDL_DEST,'VFC');  
0582     FDLSC_BN2 :       WRITE (FDL_DEST,'bin2');  
0583     FDLSC_BN4 :       WRITE (FDL_DEST,'bin4');  
0584     FDLSC_BN8 :       WRITE (FDL_DEST,'bin8');  
0585     FDLSC_PAC :       WRITE (FDL_DEST,'decimal');  
0586     FDLSC_IN2 :       WRITE (FDL_DEST,'int2');  
0587     FDLSC_IN4 :       WRITE (FDL_DEST,'int4');  
0588     FDLSC_IN8 :       WRITE (FDL_DEST,'int8');  
0589     FDLSC_STG :       WRITE (FDL_DEST,'string');  
0590     FDLSC_IAS :       WRITE (FDL_DEST,'IAS');  
0591     FDLSC_RSTS :     WRITE (FDL_DEST,'RSTS/E');  
0592     FDLSC_M :         WRITE (FDL_DEST,'RSX-11M');  
0593     FDLSC_MPLUS :    WRITE (FDL_DEST,'RSX-11M-PLUS');  
0594     FDLSC_RT :        WRITE (FDL_DEST,'RT-11');  
0595     FDLSC_TRAX :     WRITE (FDL_DEST,'TRAX-11');  
0596     FDLSC_VMS :     WRITE (FDL_DEST,'VAX/VMS');  
0597  
0598     OTHERWISE  
0599       { NULL-STATEMENT } ;  
0600  
0601   END; { CASE }  
0602  
0603 END; { SHOW_QUALIFIER }  
0604
```



```
0606      { ++
0607
0608      SHOW_CURRENT -- Display (or output) the current line_object.
0609
0610      This routine outputs the textual representation of the line_object pointed
0611      to by DEF_HEAD.
0612
0613      CALLING SEQUENCE:
0614
0615      SHOW_CURRENT (PLUS_VALUE);
0616
0617      INPUT PARAMETERS:
0618
0619      none
0620
0621      IMPLICIT INPUTS:
0622
0623      DEF_CURRENT
0624      DEST_IS_TERMINAL
0625      LINES_PER_PAGE
0626
0627      OUTPUT PARAMETERS:
0628
0629      none
0630
0631      IMPLICIT OUTPUTS:
0632
0633      FDL_DEST
0634
0635      ROUTINES CALLED:
0636
0637      SHOW_PRIMARY
0638      SHOW_SECONDARY
0639      SHOW_QUALIFIER
0640      CLEAR
0641      LIB$SIGNAL
0642
0643      ROUTINE VALUE:
0644
0645      none
0646
0647      SIGNALS:
0648
0649
0650      SIDE EFFECTS:
0651
0652      none
0653
0654      -- }
```

```
0656 PROCEDURE SHOW_CURRENT (PLUS_VALUE : BOOLEAN);
0657
0658 VAR
0659     RETLEN      : [VOLATILE]$WORD;
0660
0661     PROCEDURE SHOW_PROT (PROTECTION : CTRL_ARRAY; FIELD_OFFSET : INTEGER);
0662
0663     BEGIN
0664
0665         IF PROTECTION[FIELD_OFFSET+EDF$V_NOREAD] THEN
0666             WRITE (FDL_DEST,'R');
0667
0668         IF PROTECTION[FIELD_OFFSET+EDF$V_NOWRITE] THEN
0669             WRITE (FDL_DEST,'W');
0670
0671         IF PROTECTION[FIELD_OFFSET+EDF$V_NOEXE] THEN
0672             WRITE (FDL_DEST,'E');
0673
0674         IF PROTECTION[FIELD_OFFSET+EDF$V_NOEXE] THEN
0675             WRITE (FDL_DEST,'E');
0676
0677         IF PROTECTION[FIELD_OFFSET+EDF$V_NOEXE] THEN
0678             WRITE (FDL_DEST,'D');
0679
0680     END;      { SHOW_PROT }
0681
0682 BEGIN
0683
0684     { +
0685     If it's going to the terminal, shift it.
0686     - }
0687     IF DEST_IS_TERMINAL THEN
0688         WRITE (FDL_DEST,SHIFT);
0689
0690     { +
0691     Write to FDL_DEST, according to the type of object it is.
0692     - }
0693     CASE DEF_CURRENT^.OBJECT_TYPE OF
0694
0695         PRI :
0696
0697         WITH DEF_CURRENT^ DO
0698
0699             BEGIN
0700
0701                 { +
0702                 Output the primary.
0703                 - }
0704                 SHOW_PRIMARY;
0705
0706                 { +
0707                 Show only the primary if wanted.
0708                 - }
0709                 IF PLUS_VALUE THEN
0710
0711
0712
```

```
0713 { +
0714 If it has an end-of-line comment on it, output that also.
0715 - }
0716 IF (COMMENT.DSC$W_LENGTH > 0) THEN
0717     WRITE (FDL_DEST, ' ', COMMENT.DSC$A_POINTER^:
0718           COMMENT.DSC$W_LENGTH);
0719
0720
0721 { +
0722 This actually does the QIO.
0723 - }
0724 WRITELN (FDL_DEST);
0725
0726 END; { PRI }
0727
0728 SEC :
0729
0730 WITH DEF_CURRENT^ DO
0731 BEGIN
0732     { +
0733     Output the secondary.
0734     - }
0735     SHOW_SECONDARY;
0736
0737     { +
0738     Don't show the value unless wanted.
0739     - }
0740     IF PLUS_VALUE THEN
0741     BEGIN
0742         { +
0743         If it's a string valued secondary, put the string out.
0744         - }
0745         IF SEC_TYPE[SECONDARY].STR THEN
0746         BEGIN
0747             CHECK_QUOTES (STRING);
0748             IF (STRING.DSC$W_LENGTH > 0) THEN
0749             WRITE (FDL_DEST, ' ', STRING.DSC$A_POINTER^:
0750                   STRING.DSC$W_LENGTH);
0751             END;
0752         { +
0753         If it's a qualifier valued secondary, put the qualifier out.
0754         - }
0755         IF SEC_TYPE[SECONDARY].QUAL THEN
0756             SHOW_QUALIFIER;
0757         { +
0758
0759
0760
0761
0762
0763
0764
0765
0766
0767
0768
0769
```

Source Listing

```
0770 If it's a number valued secondary, put the number out.
0771 - )
0772 IF SEC_TYPE[SECONDARY].NUM THEN
0773     { +
0774     Show the number.
0775     - )
0776     WRITE (FDL_DEST,' ',NUMBER:NUM_LEN(NUMBER));
0777
0778 { +
0779 If it's a boolean valued secondary, put YES or NO out.
0780 - )
0781 IF SEC_TYPE[SECONDARY].SW THEN
0782     IF SWITCH THEN
0783         WRITE (FDL_DEST,' yes')
0784     ELSE
0785         WRITE (FDL_DEST,' no');
0786
0787 IF (
0788 (SECONDARY = NULL_VALUE)
0789 OR
0790 (SECONDARY = MT_PROTECTION)
0791 ) THEN
0792 BEGIN
0793     IF (
0794 (NUMBER < %X20)    ( SPACE )
0795 OR
0796 (NUMBER > %X7E)    ( ~ )
0797 ) THEN
0798     WRITE (FDL_DEST,' ',NUMBER:NUM_LEN(NUMBER))
0799 ELSE
0800     WRITE (FDL_DEST,' ',CHR(NUMBER),' ');
0801
0802 END;    ( IF SECONDARY = NULL_VALUE OR MT_PROTECTION )
0803
0804 { +
0805 Area Position is a special case.
0806 - )
0807 IF SECONDARY = POSITIONS THEN
0808 BEGIN
0809     SHOW_QUALIFIER;
0810     IF (
0811 QUALIFIER IN [ FDL$C_CYLPOS, FDL$C_LOGPOS, FDL$C_VIRPOS, FDL$C_CLUSPOS ]
0812 ) THEN
```

0770
0771
0772
0773
0774
0775
0776
0777
0778
0779
0780
0781
0782
0783
0784
0785
0786
0787
0788
0789
0790
0791
0792
0793
0794
0795
0796
0797
0798
0799
0800
0801
0802
0803
0804
0805
0806
0807
0808
0809
0810
0811
0812
0813
0814
0815
0816
0817
0818
0819
0820
0821
0822
0823
0824
0825
0826

0884
0885
0886
0887
0888
0889
0890
0891
0892
0893
0894
0895
0896
0897
0898
0899
0900
0901
0902
0903
0904
0905
0906
0907
0908
0909
0910
0911
0912
0913
0914
0915
0916
0917
0918
0919
0920
0921
0922
0923
0924
0925
0926
0927
0928
0929
0930
0931
0932
0933
0934
0935
0936
0937
0938
0939
0940

```
Source Listing
END:
{ +
PROTECTION is also a special case.
- }
IF (SECONDARY = PROTECTION) THEN
BEGIN
    WRITE (FDL_DEST, ' (system:');
    SHOW PROT (PROT_MASK,EDF$V_SYS);
    WRITE (FDL_DEST, ' owner:');
    SHOW PROT (PROT_MASK,EDF$V_OWN);
    WRITE (FDL_DEST, ' group:');
    SHOW PROT (PROT_MASK,EDF$V_GRP);
    WRITE (FDL_DEST, ' world:');
    SHOW PROT (PROT_MASK,EDF$V_WLD);
    WRITE (FDL_DEST, ')');
END; { IF TRUE SECONDARY = PROTECTION }
{ +
If it has an end-of-line comment, put that out too.
- }
IF (COMMENT.DSC$W_LENGTH > 0) THEN
    WRITE (FDL_DEST, ' ', COMMENT.DSC$A_POINTER^:
    COMMENT.DSC$W_LENGTH);
{ +
This actually does the QIO.
- }
WRITELN (FDL_DEST);
END; { IF PLUS_VALUE }
END; { SEC }
COM :
WITH DEF_CURRENT^ DO
BEGIN
    { +
    This is a full-line comment. Just output it.
    - }
    IF (COMMENT.DSC$W_LENGTH > 0) THEN
        WRITELN (FDL_DEST, COMMENT.DSC$A_POINTER^:
        COMMENT.DSC$W_LENGTH);
END; { COM }
OTHERWISE
{ NULL-STATEMENT } ;
```

```
0941
0942     END;           { CASE }
0943
0944     { +
0945     Keep track of the number of lines shown.
0946     - }
0947     LINES_SHOWN := LINES_SHOWN + 1;
0948
0949     IF DEST_IS_TERMINAL THEN
0950
0951     BEGIN
0952
0953         IF DEF_CURRENT^.FORE <> NIL THEN
0954
0955             IF (
0956                 (((LINES_PER_PAGE - EDF$C_HEADROOM) - LINES_SHOWN) < 4)
0957                 AND
0958                 ((DEF_CURRENT^.PRIMARY <> DEF_CURRENT^.FORE^.PRIMARY)
0959                 OR
0960                 (DEF_CURRENT^.PRINUM <> DEF_CURRENT^.FORE^.PRINUM))
0961             ) THEN
0962
0963             BEGIN
0964
0965                 { +
0966                 We're about to output a short primary, reset and
0967                 clear the screen (after the user hits RETURN).
0968                 - }
0969                 LINES_SHOWN := 0;
0970                 CLEAR(PAUSE);
0971
0972             END;
0973
0974             IF (
0975                 (LINES_SHOWN >= (LINES_PER_PAGE - EDF$C_HEADROOM))
0976             ) THEN
0977
0978             BEGIN
0979
0980                 { +
0981                 The counter tripped, reset and clear the screen
0982                 (after the user hits RETURN).
0983                 - }
0984                 LINES_SHOWN := 0;
0985                 CLEAR(PAUSE);
0986
0987             END;
0988
0989     END;           { IF DEST_IS_TERMINAL }
0990
0991 END;           { SHOW_CURRENT }
```

```
0993      ( ++
0994
0995      GENERATE_FDL -- Routine to output the FDL definition.
0996
0997      This routine outputs the FDL definition.
0998
0999      CALLING SEQUENCE:
1000
1001      GENERATE_FDL;
1002
1003      INPUT PARAMETERS:
1004
1005      none
1006
1007      IMPLICIT INPUTS:
1008
1009      DEF_CURRENT
1010      ANST_REVERSE
1011
1012      OUTPUT PARAMETERS:
1013
1014      none
1015
1016      IMPLICIT OUTPUTS:
1017
1018      FDL_DEST
1019      LINES_SHOWN
1020
1021      ROUTINES CALLED:
1022
1023      CLEAR
1024      SHOW_CURRENT
1025      INCR_CURRENT
1026      LIB$SIGNAL
1027
1028      ROUTINE VALUE:
1029
1030      none
1031
1032      SIGNALS:
1033
1034
1035      SIDE EFFECTS:
1036
1037      none
1038
1039      -- )
```



```
1041 PROCEDURE GENERATE_FDL;
1042
1043 VAR
1044     PREV_PRIMARY      : PRIMARY_TYPE;
1045     PREV_PRINUM      : INTEGER;
1046
1047 BEGIN
1048
1049     { +
1050     Do the Primaries as stored in the Definition Linked List.
1051     - }
1052     DEF_CURRENT := DEF_HEAD;
1053
1054     { +
1055     Setup to keep track of new primaries.
1056     - }
1057     PREV_PRINUM := -1;
1058
1059     { +
1060     Initialize the line counter (incremented in SHOW_CURRENT).
1061     - }
1062     LINES_SHOWN := 0;
1063
1064     { +
1065     Do it, if there is something to show.
1066     - }
1067     IF DEF_CURRENT <> NIL THEN
1068
1069     BEGIN
1070
1071     { +
1072     List isn't empty, cycle through the line_objects and show them.
1073     At least until the end of the list, or until the user types ^Z.
1074     - }
1075
1076     REPEAT
1077
1078     { +
1079     Produce the textual version of the Definition Linked List.
1080     Precede the 1st occurrence of a primary with 2 blank lines.
1081     - }
1082     WITH DEF_CURRENT^ DO
1083
1084     BEGIN
1085
1086     IF PREV_PRINUM < 0 THEN
1087
1088     BEGIN
1089
1090     { +
1091     Initial primary, don't skip lines here, just setup.
1092     - }
1093     PREV_PRIMARY      := PRIMARY;
1094     PREV_PRINUM      := PRINUM;
1095
1096     END
1097
```

```
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133

ELSE
BEGIN
    { +
    Skip a line if this is a new primary.
    - }
    IF NOT (
    (PREV_PRIMARY = PRIMARY) AND (PREV_PRINUM = PRINUM)
    ) THEN
        BEGIN;
            PREV_PRIMARY := PRIMARY;
            PREV_PRINUM := PRINUM;

            { +
            This is a new primary, put out a blank line.
            - }
            WRITELN (FDL_DEST);
            LINES_SHOWN := LINES_SHOWN + 1;
        END;
    END; { IF FALSE PREV_PRINUM >= 0 }
END; { WITH DEF_CURRENT^ }
SHOW_CURRENT (TRUE);
INCR_CURRENT;
UNTIL DEF_CURRENT = NIL;
END; { IF FALSE DEF_CURRENT = NIL }
END; { GENERATE_FDL }
```

```
1135      ( ++
1136
1137      VIEW_DEF -- Routine to show the user the Definition Linked List.
1138
1139      This routine will display the definition on the user's terminal.
1140
1141      CALLING SEQUENCE:
1142
1143      VIEW_DEF:
1144
1145      INPUT PARAMETERS:
1146
1147      none
1148
1149      IMPLICIT INPUTS:
1150
1151      SYSS$OUTPUT_NAME
1152      DEF_CURRENT
1153      CONTROL_ZEE_TYPED
1154
1155      OUTPUT PARAMETERS:
1156
1157      none
1158
1159      IMPLICIT OUTPUTS:
1160
1161      SYSS$INPUT: the terminal
1162      DEST_IS_TERMINAL
1163
1164      ROUTINES CALLED:
1165
1166      CLEAR
1167      GENERATE_FDL
1168
1169      ROUTINE VALUE:
1170
1171      none
1172
1173      SIGNALS:
1174
1175      none
1176
1177      SIDE EFFECTS:
1178
1179      none
1180
1181      -- }
```

```
1183 PROCEDURE VIEW_DEF;
1184
1185 BEGIN
1186     { +
1187     Erase the user's screen.
1188     - }
1189     CLEAR (SCREEN);
1190
1191     { +
1192     'Open' his terminal and initialize it.
1193     Close it first to avoid conflicts.
1194     - }
1195     DEST_IS_TERMINAL := TRUE;
1196
1197     CLOSE      (FDL_DEST,ERROR := CONTINUE);
1198     OPEN       (FDL_DEST,SYSS$OUTPUT_NAME,NEW,RECORD_LENGTH := 252);
1199     REWRITE   (FDL_DEST);
1200
1201     { +
1202     Put the current definition out to the terminal.
1203     - }
1204     GENERATE_FDL;
1205
1206     { +
1207     We're done, close it off.
1208     - }
1209     CLOSE      (FDL_DEST);
1210
1211     { +
1212     Don't clear if the user hit ^Z, or if SHOW_CURRENT had just done a PAUSE.
1213     - }
1214     IF (NOT CONTROL_ZEE_TYPED) AND (LINES_SHOWN <> 0) THEN
1215     BEGIN
1216
1217         WRITELN;
1218         CLEAR (PAUSE);
1219
1220     END;
1221
1222 END; { VIEW_DEF }
```

```
1226 { ++
1227
1228 SHOW_PRIMARY_SECTION -- Display the whole primary section.
1229
1230 This routine outputs the selected primary section to the screen.
1231
1232 CALLING SEQUENCE:
1233
1234 SHOW_PRIMARY_SECTION (TEST);
1235
1236 INPUT PARAMETERS:
1237
1238 TEST
1239
1240 IMPLICIT INPUTS:
1241
1242 FDL_DEST
1243 CONTROL_ZEE_TYPED
1244 DEF_CURRENT
1245 DEF_HEAD
1246
1247 OUTPUT PARAMETERS:
1248
1249 none
1250
1251 IMPLICIT OUTPUTS:
1252
1253 SYSSOUTPUT:
1254 DEF_CURRENT
1255
1256 ROUTINES CALLED:
1257
1258 CLEAR
1259 INCR_CURRENT
1260 SECTION_MATCH
1261
1262 ROUTINE VALUE:
1263
1264 none
1265
1266 SIGNALS:
1267
1268 none
1269
1270 SIDE EFFECTS:
1271
1272 none
1273
1274 -- }
```

```
1276 [GLOBAL] PROCEDURE SHOW_PRIMARY_SECTION (TEST : LINE_OBJECT);
1277
1278 VAR
1279     AT_PRIMARY : BOOLEAN;
1280
1281 BEGIN
1282
1283     { +
1284     Show him that whole primary section.
1285     - }
1286     LINES_SHOWN      := 0;
1287
1288     { +
1289     Step through the whole list.
1290     - }
1291     DEF_CURRENT      := DEF_HEAD;
1292     AT_PRIMARY       := FALSE;
1293
1294     REPEAT
1295
1296         { +
1297         If DEF_CURRENT points to a line_object in that primary, show it.
1298         - }
1299         IF CURRENT_EQ_TEST (TEST,FALSE) THEN
1300
1301             BEGIN
1302
1303                 SHOW_CURRENT (TRUE);
1304                 AT_PRIMARY := TRUE;
1305
1306             END
1307
1308         ELSE
1309
1310             BEGIN
1311
1312                 IF AT_PRIMARY THEN
1313
1314                     DEF_CURRENT := NIL;
1315
1316                 END;
1317
1318                 { +
1319                 Try the next.
1320                 - }
1321                 INCR_CURRENT;
1322
1323             UNTIL DEF_CURRENT = NIL;
1324
1325             WRITELN (FDL_DEST);
1326
1327     END; { SHOW_PRIMARY_SECTION }
```

```
1329      ( ++
1330
1331      SHOW_ALL_PRIMARIES -- Display the existing primary attributes.
1332
1333      This routine outputs the existing primaries on the screen.
1334
1335      CALLING SEQUENCE:
1336
1337      SHOW_ALL_PRIMARIES;
1338
1339      INPUT PARAMETERS:
1340
1341      none
1342
1343      IMPLICIT INPUTS:
1344
1345      FDL_DEST
1346      CONTROL_ZEE_TYPED
1347      DEF_CURRENT
1348      DEF_HEAD
1349
1350      OUTPUT PARAMETERS:
1351
1352      none
1353
1354      IMPLICIT OUTPUTS:
1355
1356      SYS$OUTPUT:
1357      DEF_CURRENT
1358
1359      ROUTINES CALLED:
1360
1361      CLEAR
1362      INCR_CURRENT
1363      SECTION_MATCH
1364
1365      ROUTINE VALUE:
1366
1367      none
1368
1369      SIGNALS:
1370
1371      none
1372
1373      SIDE EFFECTS:
1374
1375      none
1376
1377      -- }
```

```
1379 [GLOBAL] PROCEDURE SHOW_ALL_PRIMARIES;
1380
1381 BEGIN
1382
1383   { +
1384   Show him all the primary attributes.
1385   - }
1386   LINES_SHOWN      := 0;
1387
1388   { +
1389   Step through the whole list.
1390   - }
1391   DEF_CURRENT      := DEF_HEAD;
1392
1393   REPEAT
1394
1395     { +
1396     If DEF_CURRENT points to a line_object that is a primary, show it.
1397     But not the Ident.
1398     - }
1399     IF (
1400     (DEF_CURRENT^.OBJECT_TYPE = PRI)
1401     AND
1402     (DEF_CURRENT^.PRIMARY <> IDENT)
1403     ) THEN
1404
1405       SHOW_CURRENT (TRUE);
1406
1407     { +
1408     Try the next.
1409     - }
1410     INCR_CURRENT;
1411
1412   UNTIL DEF_CURRENT = NIL;
1413
1414   WRITELN (FDL_DEST);
1415
1416 END; { SHOW_ALL_PRIMARIES }
```



```
1418 { ++
1419
1420 SHOW_CUR_PRI_SEC -- Display the line_object pointed to by def_current.
1421
1422 This routine outputs the def_current line_object.
1423
1424 CALLING SEQUENCE:
1425
1426 SHOW_CUR_PRI_SEC;
1427
1428 INPUT PARAMETERS:
1429
1430 none
1431
1432 IMPLICIT INPUTS:
1433
1434 FDL_DEST
1435 CONTROL_ZEE_TYPED
1436 DEF_CURRENT
1437 DEF_HEAD
1438
1439 OUTPUT PARAMETERS:
1440
1441 none
1442
1443 IMPLICIT OUTPUTS:
1444
1445 SYSSOUTPUT:
1446 DEF_CURRENT
1447
1448 ROUTINES CALLED:
1449
1450 CLEAR
1451 INCR_CURRENT
1452 SECTION_MATCH
1453
1454 ROUTINE VALUE:
1455
1456 none
1457
1458 SIGNALS:
1459
1460 none
1461
1462 SIDE EFFECTS:
1463
1464 none
1465
1466 -- )
```

```
1468 [GLOBAL] PROCEDURE SHOW_CUR_PRI_SEC (PLUS_VALUE : BOOLEAN);
1469
1470 VAR
1471     SAVE_OBJECT_TYPE           : LINE_OBJECT_TYPE;
1472
1473 BEGIN
1474     LINES_SHOWN                := 0;
1475
1476     { +
1477     Now display the 2 versions of the line_object.
1478     All this only works because the primary that a line_object is in
1479     is stored redundantly in all the subsequent secondaries.
1480     - }
1481     SAVE_OBJECT_TYPE           := DEF_CURRENT^.OBJECT_TYPE;
1482
1483     DEF_CURRENT^.OBJECT_TYPE   := PRI;
1484     SHOW_CURRENT (PLUS_VALUE);
1485
1486     IF DEF_CURRENT^.PRIMARY <> TITLE THEN
1487     BEGIN
1488         DEF_CURRENT^.OBJECT_TYPE := SEC;
1489         SHOW_CURRENT (PLUS_VALUE);
1490     END;
1491
1492     DEF_CURRENT^.OBJECT_TYPE   := SAVE_OBJECT_TYPE;
1493
1494 END; { SHOW_CUR_PRI_SEC }
1495
1496 END.
1497
1498 { End of file: SRC$:EDFSHOW.PAS }
1499
1500
1501
```

Generated Code													.TITLE	EDFSHOW		
													.IDENT	\V04-000\		
													00000	.PSECT	\$CODE,PIC,CON,REL,LCL,SHR,EXE,RD,NOWRT,2	
0000006B	00000064	00000055	00000010	00000048	00000	C.AAA:	.LONG	72,16,85,100,107,111,128,144,149,157,162,-	00014	168,174,182,186,194,202,209	00028			0003C		
0000009D	00000095	00000090	00000080	0000006F	00048		.ASCII	<12>\PRIMARY_TYPE\	00055					00063		
000000BA	000000B6	000000AE	000000A8	000000A2	00064		.ASCII	<14>\DUMMY_PRIMARYS\	0006B					0006F		
59 45 50 59	54 5F 59	52 41	4D 49 52	50 0C	00063		.ASCII	<6>\ACCESS\	00064					0006B		
59 52 41 4D	49 52 50	5F 59	4D 4D 55	44 0E	00064		.ASCII	<3>\ACL\	0006B					0006F		
					00063		.ASCII	<16>\ANALYSIS_OF_AREA\	0006F					0007D		
					00064		.ASCII	<15>\ANALYSIS_OF_KEY\	0007D					00080		
					00064		.ASCII	<4>\AREA\	0007D					00080		
					00064		.ASCII	<7>\CONNECT\	00080					0008E		
					00064		.ASCII	<4>\DATE\	0008E					00090		
					00064		.ASCII	<5>\FILES\	00090					00095		
					00064		.ASCII	<5>\IDENT\	00095					0009D		
					00064		.ASCII	<7>\JOURNAL\	0009D					000A2		
					00064		.ASCII	<3>\KEY\	000A2					000A8		
					00064		.ASCII	<7>\RECORDS\	000A8					000AE		
					00064		.ASCII	<7>\SHARING\	000AE					000B6		
					00064		.ASCII	<6>\SYSTEM\	000B6					000BA		
					00064		.ASCII	<5>\TITLE\<0>	000BA					000C2		
					00064		.BYTE	^X38,8	000C2					000CA		
					00064		.BYTE	0,^X82	000CA					000D1		
					00064		.LONG	0,0,0,0	000D1					000DB		
					00064		.BYTE	^XE0,0,0	000DB					000DA		
					00064		.BLKB	1	000DA					000DC		
					00064		.LONG	612,151,627,644,654,662,667,672,683,693,-	000DC					000EC		
					00064		.LONG	701,707,723,734,756,781,799,819,829,835,-	000EC					000EF		
					00064		.LONG	855,874,886,907,927,944,962,978,993,1013,-	000EF					000F0		
					00064		.LONG	1025,1046,1059,1071,1090,1101,1111,1119,-	000F0					00104		
					00064		.LONG	1132,1141,1153,1161,1173,1186,1198,1215,-	00104					00118		
					00064		.LONG	1233,1250,1260,1272,1285,1299,1316,1333,-	00118					0012C		
					00064		.LONG	1351,1358,1377,1388,1404,1419,1434,1450,-	0012C					00140		
					00064		.LONG	1470,1486,1496,1516,1531,1548,1558,1574,-	00140					00154		
					00064		.LONG	1587,1595,1605,1617,1627,1638,1658,1670,-	00154					00168		
					00064		.LONG	1683,1692,1703,1713,1726,1741,1757,1773,-	00168					0017C		
					00064		.LONG	1789,1799,1819,1833,1853,1864,1878,1893,-	0017C					00190		
					00064		.LONG	1909,1927,1944,1949,1958,1978,1996,2009,-	00190					001A4		
					00064		.LONG	2015,2030,2041,2052,2061,2077,2093,2103,-	001A4					001B8		
					00064		.LONG	2113,2131,2146,2158,2170,2182,2193,2205,-	001B8					001CC		
					00064		.LONG	2216,2229,2241,2255,2263,2273,2283,2304,-	001CC					001E0		
					00064		.LONG	2328,2339,2350,2368,2379,2397,2403,2412,-	001E0					001F4		
					00064		.LONG	2423,2432,2443,2456	001F4					00208		
					00064		.LONG		00208					0021C		
					00064		.LONG		0021C					00230		
					00064		.LONG		00230					00244		
					00064		.LONG		00244					00258		
					00064		.LONG		00258					0026C		
					00064		.LONG		0026C					00280		
					00064		.LONG		00280					00294		
					00064		.LONG		00294					002AB		

	000008B5	000008A8	0000089D	00000891	00000886	002BC										
	000008EB	000008E1	000008D7	000008CF	000008C1	002D0										
	00000940	0000092E	00000923	00000918	00000900	002E4										
	00000977	0000096C	00000963	0000095D	0000094B	002F8										
				0000098B	00000980	0030C										
	000009D7	000009D0	000009BD	000009AC	000009A1	00314	.LONG 2465,2476,2493,2512,2519,2524,2531,2535,-									
	000009FC	000009F3	000009E7	000009E3	000009DC	00328	2547,2556,2560,2567,2582,2589,2596									
50	59	54	5F	59	52	41	44	4E	4F	43	45	53	0E	0033C		
													45	00350	.ASCII <14>\SECONDARY_TYPE\	
41	44	4E	4F	43	45	53	5F	59	4D	4D	55	44	10	0035E		
													24	0035F	.ASCII <16>\DUMMY_SECONDARY\$\	
				24	4F	49	5F	4B	43	4F	4C	42	09	0036D		
						24	45	54	45	4C	45	44	07	00370	.ASCII <9>\BLOCK_IOS\	
									24	54	45	47	04	0037A	.ASCII <7>\DELETES\	
									24	54	55	50	04	00382	.ASCII <4>\GETS\	
									24	54	55	50	04	00387	.ASCII <4>\PUTS\	
	24	4F	49	5F	44	52	4F	43	45	52	0A			0038C	.ASCII <10>\RECORD_IOS\	
		24	45	54	41	43	4E	55	52	54	09			00397	.ASCII <9>\TRUNCATES\	
				24	45	54	41	44	50	55	07			003A1	.ASCII <7>\UPDATES\	
							59	52	54	4E	45	05		003A9	.ASCII <5>\ENTRY\	
41	50	53	5F	44	45	4D	49	41	4C	43	45	52	0F	003AF	.ASCII <15>\RECLAIMED_SPACE\	
												45	43	003BD		
				24	4C	4C	49	46	5F	41	54	41	44	0A	003BF	.ASCII <10>\DATA_FILLS\
50	4D	4F	43	5F	59	45	4B	5F	41	54	41	44	15	003CA	.ASCII <21>\DATA_KEY_COMPRESSIONS\	
						24	4E	4F	49	53	53	45	52	003DB		
43	5F	44	52	4F	43	45	52	5F	41	54	41	44	18	003E0	.ASCII <24>\DATA_RECORD_COMPRESSIONS\	
			24	4E	4F	49	53	53	45	52	50	4D	4F	003EE		
43	5F	44	52	4F	43	45	52	5F	41	54	41	44	11	003F9	.ASCII <17>\DATA_RECORD_COUNT\	
										54	4E	55	4F	00407		
43	4F	5F	45	43	41	50	53	5F	41	54	41	44	13	0040B	.ASCII <19>\DATA_SPACE_OCCUPIED\	
								44	45	49	50	55	43	00419		
				53	4E	4F	49	54	45	4C	45	44	09	0041F	.ASCII <9>\DELETIONS\	
								48	54	50	45	44	05	00429	.ASCII <5>\DEPTH\	
45	50	5F	53	45	54	41	43	49	4C	50	55	44	13	0042F	.ASCII <19>\DUPLICATES_PER_SIDR\	
								52	44	49	53	5F	52	0043D		
53	45	52	50	4D	4F	43	5F	58	45	44	4E	49	12	00443	.ASCII <18>\INDEX_COMPRESSIONS\	
								24	4E	4F	49	53		00451		
		24	4C	4C	49	46	5F	58	45	44	4E	49	08	00456	.ASCII <11>\INDEX_FILLS\	
4F	5F	45	43	41	50	53	5F	58	45	44	4E	49	14	00462	.ASCII <20>\INDEX_SPACE_OCCUPIED\	
							44	45	49	50	55	43	43	00470		
44	52	4F	43	45	52	5F	31	4C	45	56	45	4C	13	00477	.ASCII <19>\LEVEL1_RECORD_COUNT\	
								54	4E	55	4F	43	5F	00485		
4E	45	4C	5F	41	54	41	44	5F	4E	41	45	4D	10	0048B	.ASCII <16>\MEAN_DATA_LENGTH\	
												48	54	00499		
45	4C	5F	58	45	44	4E	49	5F	4E	41	45	4D	11	0049C	.ASCII <17>\MEAN_INDEX_LENGTH\	
										48	54	47	4E	004AA		
53	53	45	43	43	41	5F	4D	4F	44	4E	41	52	0F	004AE	.ASCII <15>\RANDOM_ACCESSES\	
												53	45	004BC		
54	52	45	53	4E	49	5F	4D	4F	44	4E	41	52	0E	004BE	.ASCII <14>\RANDOM_INSERTS\	
													53	004CC		
43	41	5F	4C	41	49	54	4E	45	55	51	45	53	13	004CD	.ASCII <19>\SEQUENTIAL_ACCESSES\	
								53	45	53	53	45	43	004DB		
		24	4E	4F	49	54	41	43	4F	4C	4C	41	0B	004E1	.ASCII <11>\ALLOCATIONS\	
54	4E	4F	43	5F	59	52	54	5F	54	53	45	42	14	004ED	.ASCII <20>\BEST_TRY_CONTIGUOUS\	
						24	53	55	4F	55	47	49		004FB		
	24	45	5A	49	53	5F	54	45	4B	43	55	42	0C	00502	.ASCII <12>\BUCKET_SIZES\	
		24	53	55	4F	55	47	49	54	4E	4F	43	0B	0050F	.ASCII <11>\CONTIGUOUS\	

Generated Code

4F	49	54	49	53	4F	50	5F	54	43	41	58	45	12
			24	4E	4F	49	53	4E	45	54	58	45	4E
			24	4E	4F	49	54	49	53	4F	50	09	
					24	45	4D	55	4C	4F	56	07	
	53	55	4F	4E	4F	52	48	43	4E	59	53	41	0C
					4F	49	5F	4B	43	4F	4C	42	08
		45	44	4F	43	5F	54	45	4B	43	55	42	0B
						54	58	45	54	4E	4F	43	07
		45	4C	49	46	5F	46	4F	5F	44	4E	45	0B
	53	54	45	4B	43	55	42	5F	4C	4C	49	46	0C
		45	54	45	4C	45	44	5F	54	53	41	46	0B
45	52	45	46	45	52	5F	46	4F	5F	59	45	4B	10
											45	43	4E
45	5F	52	45	54	41	45	52	47	5F	59	45	4B	11
											4C	41	55
54	5F	52	45	54	41	45	52	47	5F	59	45	4B	10
											4E	41	48
				54	49	4D	49	4C	5F	59	45	4B	09
		45	44	4F	4D	5F	45	54	41	43	4F	4C	0B
45	44	41	45	52	5F	4E	4F	5F	4B	43	4F	4C	0C
45	54	49	52	57	5F	4E	4F	5F	4B	43	4F	4C	0D
4B	43	4F	4C	4E	55	5F	4C	41	55	4E	41	4D	10
											47	4E	49
4F	43	5F	4B	43	4F	4C	42	49	54	4C	55	4D	10
											54	4E	55
43	5F	52	45	46	46	55	42	49	54	4C	55	4D	11
											54	4E	55
											4E	55	4F
											4E	06	
52	5F	54	4E	45	54	53	49	58	45	4E	4F	4E	12
											44	52	4F
											44	52	4F
											44	52	4F
45	4C	44	52	41	47	45	52	5F	44	41	45	52	0A
											52	0F	
											53	53	
4C	42	41	4E	45	5F	54	55	4F	45	4D	49	54	0E
													45
4F	49	52	45	50	5F	54	55	4F	45	4D	49	54	0E
													44
50	5F	4E	4F	5F	45	54	41	43	4E	55	52	54	0F
													54
4E	4F	43	5F	4C	45	43	4E	41	43	5F	54	54	13
											4F	5F	4C
50	4E	49	5F	45	53	41	43	50	55	5F	54	54	0F
													54
													55
													09
45	50	59	54	5F	45	47	52	55	50	5F	54	54	13
													54
													13
48	43	45	4F	4E	5F	44	41	45	52	5F	54	54	0E
													4F
4C	49	46	4F	4E	5F	44	41	45	52	5F	54	54	10
													52
													45
													54
													09
4F	43	45	52	5F	52	4F	46	5F	54	49	41	57	0F
													44
													52
		44	4E	49	48	45	42	5F	45	54	49	52	57
													0C
													44
													07
													09

0051B	.ASCII	<18>\EXACT_POSITIONINGS\
00529		
0052E	.ASCII	<10>\EXTENSIONS\
00539	.ASCII	<9>\POSITIONS\
00543	.ASCII	<7>\VOLUMES\
0054B	.ASCII	<12>\ASYNCHRONOUS\
00558	.ASCII	<8>\BLOCK_IO\
00561	.ASCII	<11>\BUCKET_CODE\
0056D	.ASCII	<7>\CONTEXT\
00575	.ASCII	<11>\END_OF_FILE\
00581	.ASCII	<12>\FILE_BUCKETS\
0058E	.ASCII	<11>\FAST_DELETE\
0059A	.ASCII	<16>\KEY_OF_REFERENCE\
005A8		
005AB	.ASCII	<17>\KEY_GREATER_EQUAL\
005B9		
005BD	.ASCII	<16>\KEY_GREATER_THAN\
005CB		
005CE	.ASCII	<9>\KEY_LIMIT\
005D8	.ASCII	<11>\LOCATE_MODE\
005E4	.ASCII	<12>\LOCK_ON_READ\
005F1	.ASCII	<13>\LOCK_ON_WRITE\
005FF	.ASCII	<16>\MANUAL_UNLOCKING\
0060D		
00610	.ASCII	<16>\MULTIBLOCK_COUNT\
0061E		
00621	.ASCII	<17>\MULTIBUFFER_COUNT\
0062F		
00633	.ASCII	<6>\NOLOCK\
0063A	.ASCII	<18>\NONEXISTENT_RECORD\
00648		
0064D	.ASCII	<10>\READ_AHEAD\
00658	.ASCII	<15>\READ_REGARDLESS\
00666		
00668	.ASCII	<14>\TIMEOUT_ENABLE\
00676		
00677	.ASCII	<14>\TIMEOUT_PERIOD\
00685		
00686	.ASCII	<15>\TRUNCATE_ON_PUT\
00694		
00696	.ASCII	<19>\TT_CANCEL_CONTROL_0\
006A4		
006AA	.ASCII	<15>\TT_UPCASE_INPUT\
006B8		
006BA	.ASCII	<9>\TT_PROMPT\
006C4	.ASCII	<19>\TT_PURGE_TYPE_AHEAD\
006D2		
006D8	.ASCII	<14>\TT_READ_NOECHO\
006E6		
006E7	.ASCII	<16>\TT_READ_NOFILTER\
006F5		
006F8	.ASCII	<9>\UPDATE_IF\
00702	.ASCII	<15>\WAIT_FOR_RECORD\
00710		
00712	.ASCII	<12>\WRITE_BEHIND\
0071F	.ASCII	<7>\BACKUPS\
00727	.ASCII	<9>\CREATIONS\

Generated Code

		24	4E	4F	49	54	41	52	49	50	58	45	0B	00731	.ASCII	<11>\EXPIRATIONS\	
				24	4E	4F	49	53	49	56	45	52	09	0073D	.ASCII	<9>\REVISIONS\	
			4E	4F	49	54	41	43	4F	4C	4C	41	0A	00747	.ASCII	<10>\ALLOCATION\	
54	4E	4F	43	5F	59	52	54	5F	54	53	45	42	13	00752	.ASCII	<19>\BEST_TRY_CONTIGUOUS\	
								53	55	4F	55	47	49	00760			
		45	5A	49	53	5F	54	45	4B	43	55	42	0B	00766	.ASCII	<11>\BUCKET_SIZE\	
	45	5A	49	53	5F	52	45	54	53	55	4C	43	0C	00772	.ASCII	<12>\CLUSTER_SIZE\	
					24	54	58	45	54	4E	4F	43	08	0077F	.ASCII	<8>\CONTEXTS\	
				53	55	4F	55	47	49	54	4E	4F	43	0A	00788	.ASCII	<10>\CONTIGUOUS\
				46	49	5F	45	54	41	45	52	43	09	00793	.ASCII	<9>\CREATE_IF\	
54	45	4D	41	4E	5F	54	4C	55	41	46	45	44	0C	0079D	.ASCII	<12>\DEFAULT_NAME\	
	49	52	57	5F	44	45	52	52	45	46	45	44	0E	007AA	.ASCII	<14>\DEFERRED_WRITE\	
													45	007B8			
4F	4C	43	5F	4E	4F	5F	45	54	45	4C	45	44	0F	007B9	.ASCII	<15>\DELETE_ON_CLOSE\	
													45	007C7			
54	4E	45	5F	59	52	4F	54	43	45	52	49	44	0F	007C9	.ASCII	<15>\DIRECTORY_ENTRY\	
													59	007D7			
45	4C	45	44	5F	4E	4F	5F	45	53	41	52	45	0F	007D9	.ASCII	<15>\ERASE_ON_DELETE\	
													45	007E7			
				4E	4F	49	53	4E	45	54	58	45	09	007E9	.ASCII	<9>\EXTENSION\	
52	45	46	46	55	42	5F	4C	41	42	4F	4C	47	13	007F3	.ASCII	<19>\GLOBAL_BUFFER_COUNT\	
								54	4E	55	4F	43	5F	00801			
45	5A	49	53	5F	4B	43	4F	4C	42	5F	54	4D	0D	00807	.ASCII	<13>\MT_BLOCK_SIZE\	
4F	50	5F	54	4E	45	52	52	55	43	5F	54	4D	13	00815	.ASCII	<19>\MT_CURRENT_POSITION\	
								4E	4F	49	54	49	53	00823			
4E	4F	49	54	43	45	54	4F	4E	5F	54	4D	0A	00829	.ASCII	<10>\MT_NOT_EOF\		
4E	49	57	45	52	5F	4E	45	50	4F	5F	54	4D	0D	00834	.ASCII	<13>\MT_PROTECTION\	
													0E	00842	.ASCII	<14>\MT_OPEN_REWIND\	
													44	00850			
49	57	45	52	5F	45	53	4F	4C	43	5F	54	4D	0F	00851	.ASCII	<15>\MT_CLOSE_REWIND\	
													44	0085F			
55	4E	5F	44	52	4F	43	45	52	5F	58	41	4D	11	00861	.ASCII	<17>\MAX_RECORD_NUMBER\	
													45	0086F			
53	52	45	56	5F	45	5A	49	4D	49	58	41	4D	10	00873	.ASCII	<16>\MAXIMIZE_VERSION\	
													4E	00881			
									45	4D	41	4E	04	00884	.ASCII	<4>\NAME\	
				50	55	4B	43	41	42	4F	4E	08	00889	.ASCII	<8>\NOBACKUP\		
55	52	54	53	5F	45	4C	49	46	5F	4E	4F	4E	13	00892	.ASCII	<19>\NON_FILE_STRUCTURED\	
								44	45	52	55	54	43	008A0			
50	5F	45	4C	49	46	5F	54	55	50	54	55	4F	11	008A6	.ASCII	<17>\OUTPUT_FILE_PARSE\	
										45	53	52	41	008B4			
		4E	4F	49	54	41	5A	49	4E	41	47	52	4F	0C	008B8	.ASCII	<12>\ORGANIZATION\
								52	45	4E	57	4F	05	008C5	.ASCII	<5>\OWNER\	
53	4F	4C	43	5F	4E	4F	5F	54	4E	49	52	50	0E	008CB	.ASCII	<14>\PRINT_ON_CLOSE\	
													45	008D9			
			4E	4F	49	54	43	45	54	4F	52	50	0A	008DA	.ASCII	<10>\PROTECTION\	
			4B	43	45	48	43	5F	44	41	45	52	0A	008E5	.ASCII	<10>\READ_CHECK\	
				4E	4F	49	53	49	56	45	52	08	008F0	.ASCII	<8>\REVISION\		
4E	4F	5F	4C	41	49	54	4E	45	55	51	45	53	0F	008F9	.ASCII	<15>\SEQUENTIAL_ONLY\	
													59	00907			
4F	4C	43	5F	4E	4F	5F	54	49	4D	42	55	53	0F	00909	.ASCII	<15>\SUBMIT_ON_CLOSE\	
													45	00917			
				45	44	45	53	52	45	50	55	53	09	00919	.ASCII	<9>\SUPERSEDE\	
				59	52	41	52	4F	50	4D	45	54	09	00923	.ASCII	<9>\TEMPORARY\	
43	5F	4E	4F	5F	45	54	41	43	4E	55	52	54	11	0092D	.ASCII	<17>\TRUNCATE_ON_CLOSE\	
										45	53	4F	4C	0093B			
45	50	4F	5F	45	4C	49	46	5F	52	45	53	55	0E	0093F	.ASCII	<14>\USER_FILE_OPEN\	

												4E	0094D											
		45	5A	49	53	5F	57	4F	44	4E	49	57	0B	0094E	.ASCII <11>\WINDOW_SIZE\									
		4B	43	45	48	43	5F	45	54	49	52	57	0B	0095A	.ASCII <11>\WRITE_CHECK\									
		45	47	41	4D	49	5F	52	45	54	46	41	0B	00966	.ASCII <11>\AFTER_IMAGE\									
			45	4D	41	4E	5F	52	45	54	46	41	0A	00972	.ASCII <10>\AFTER_NAME\									
		4C	49	41	52	54	5F	54	49	44	55	41	0B	0097D	.ASCII <11>\AUDIT_TRAIL\									
			45	4D	41	4E	5F	54	49	44	55	41	0A	00989	.ASCII <10>\AUDIT_NAME\									
	45	47	41	4D	49	5F	45	52	4F	46	45	42	0C	00994	.ASCII <12>\BEFORE_IMAGE\									
		45	4D	41	4E	5F	45	52	4F	46	45	42	0B	009A1	.ASCII <11>\BEFORE_NAME\									
54	49	4E	55	5F	59	52	45	56	4F	43	45	52	0D	009AD	.ASCII <13>\RECOVERY_UNIT\									
					53	45	47	4E	41	48	43	07		009BB	.ASCII <7>\CHANGES\									
				41	45	52	41	5F	41	54	41	44	09	009C3	.ASCII <9>\DATA_AREA\									
				4C	4C	49	46	5F	41	54	41	44	09	009CD	.ASCII <9>\DATA_FILL\									
50	4D	4F	43	5F	59	45	4B	5F	41	54	41	44	14	009D7	.ASCII <20>\DATA_KEY_COMPRESSION\									
							4E	4F	49	53	53	45	52	009E5										
43	5F	44	52	4F	43	45	52	5F	41	54	41	44	17	009EC	.ASCII <23>\DATA_RECORD_COMPRESSION\									
					4E	4F	49	53	53	45	52	50	4D	4F	009FA									
					53	45	54	41	43	49	4C	50	55	44	0A	00A04	.ASCII <10>\DUPLICATES\							
					41	45	52	41	5F	58	45	44	4E	49	0A	00A0F	.ASCII <10>\INDEX_AREA\							
53	45	52	50	4D	4F	43	5F	58	45	44	4E	49	11		00A1A	.ASCII <17>\INDEX_COMPRESSION\								
									4E	4F	49	53			00A28									
									4E	4F	49	53			00A2C	.ASCII <10>\INDEX_FILL\								
5F	58	45	4C	4C	49	46	5F	58	45	44	4E	49	0A		00A37	.ASCII <17>\LEVELT_INDEX_AREA\								
										41	45	52	41		00A45									
									24	45	4D	41	4E	05	00A49	.ASCII <5>\NAMES\								
									45	45	4D	41	4E	08	00A4F	.ASCII <8>\NULL_KEY\								
									45	55	4C	4C	55	4E	0A	00A58	.ASCII <10>\NULL_VALUE\							
									45	55	47	4F	4C	55	4E	0A	00A63	.ASCII <8>\PROLOGUE\						
									48	54	47	4E	45	4C	5F	47	45	53	0A	00A6C	.ASCII <10>\SEG_LENGTH\			
									4E	4F	49	54	49	53	4F	50	5F	47	45	53	0C	00A77	.ASCII <12>\SEG_POSITION\	
										45	50	59	54	5F	47	45	53	08				00A84	.ASCII <8>\SEG_TYPE\	
										4E	41	50	53	5F	4B	43	4F	4C	42	0A			00A8D	.ASCII <10>\BLOCK_SPAN\
54	4E	4F	43	5F	45	47	41	49	52	52	41	43	10		00A98	.ASCII <16>\CARRIAGE_CONTROL\								
											4C	4F	52		00AA6									
44	4C	45	49	46	5F	4C	4F	52	54	4E	4F	43	12		00AA9	.ASCII <18>\CONTROL_FIELD_SIZE\								
										45	5A	49	53	5F	00AB7									
									54	41	4D	52	4F	46	06	00ABC	.ASCII <6>\FORMAT\							
									45	5A	49	53	04		00AC3	.ASCII <4>\SIZE\								
									45	54	45	4C	45	44	06	00AC8	.ASCII <6>\DELETE\							
										54	45	47	03		00ACF	.ASCII <3>\GET\								
															00AD3	.ASCII <11>\MULTISTREAM\								
															00ADF	.ASCII <8>\PROHIBIT\								
															00AEB	.ASCII <3>\PUT\								
															00AEC	.ASCII <6>\UPDATE\								
43	4F	4C	52	45	54	4E	49	5F	52	45	53	55	0E		00AF3	.ASCII <14>\USER_INTERLOCK\								
															00B01									
															00B02	.ASCII <6>\DEVICE\								
															00B09	.ASCII <6>\SOURCE\								
															00B10	.ASCII <6>\TARGET\<0>								
															00B1C	C.AAF: .LONG ^X41,0,0,0,0,0,0,0								
															00B30									
															00B3C	C.AAG: .ASCII \ \								
															00B40	C.AAH: .ASCII <9><9><0><0>								
															00B44	C.AAI: .ASCII <9>\SEG\								
															00B48	C.AAJ: .ASCII _LENGTH\<9>								
															00B50	C.AAK: .ASCII <9>\SEG\								
															00B54	C.AAL: .ASCII _POSITION\<9><0><0>								

```

00000000 00000000 00000000 00000000 00000041
00000000 00000000 00000000
20 20 20 20
00 00 09 09
47 45 53 09
09 48 54 47 4E 45 4C 5F
47 45 53 09
00 00 09 4E 4F 49 54 49 53 4F 50 5F

```

Generated Code

```

00 72 65 64 6E 69 00 09 09 45 50 59 54 09
6C 79 63 5F 79 6E 61 09
00 72 65 74 73 75 6C 63
72 65 64 6E 69 6C 79 63
00 44 49 5F 65 6C 69 66
00 00 00 65 6D 61 6E 5F 65 6C 69 66
00 6C 61 63 69 67 6F 6C
00 6C 61 75 74 72 69 76
00 00 00 65 6E 6F 6E 09
64 65 78 65 64 6E 69 09
00 00 00 65 76 69 74 61 6C 65 72 09
00 6C 61 69 74 6E 65 75 71 65 73 09
72 65 76 6F 63 65 72 5F 6E 69 5F 66 69 09
5F 6F 74 5F 79 72 61 73 73 65 63 65 6E 09
72 75 6F 6A 5F 55 52 5F 72 65 76 65 6E 09
75 74 65 72 5F 65 67 61 69 72 72 61 63 09
6E 72
4E 41 52 54 52 4F 46 09
00 00 74 6E 69 72 70 09
00 00 64 65 78 69 66 09
00 6D 61 65 72 74 73 09
00 00 52 43 5F 6D 61 65 72 74 73 09
00 00 46 4C 5F 6D 61 65 72 74 73 09
00 00 64 65 6E 69 66 65 64 6E 75 09
00 00 00 65 6C 62 61 69 72 61 76 09
00 00 00 32 6E 69 62 09
00 00 00 34 6E 69 62 09
00 00 00 38 6E 69 62 09
6C 61 6D 69 63 65 64 09
00 00 00 32 74 6E 69 09
00 00 00 34 74 6E 69 09
00 00 00 38 74 6E 69 09
00 67 6E 69 72 74 73 09
00 45 2F 53 54 53 52 09
4D 31 31 2D 58 53 52 09
00 53 55 4C 50 2D 4D 31 31 2D 58 53 52 09
00 00 31 31 2D 54 52 09
31 31 2D 58 41 52 54 09
53 4D 56 2F 58 41 56 09
00 6F 6E 09
00 00 27 09
00000000 00000000 00000000 00000000 000000A6
00000000 00000000 00000000
00 00 00 49 25 21 5F 21
00 00 00 3A 6D 65 74 73 79 73 28 09
3A 72 65 6E 77 6F 20 2C
3A 70 75 6F 72 67 20 2C
3A 64 6C 72 6F 77 20 2C

```

```

00B60 C.AAM: .ASCII <9>\TYPE\<9><9><0>
00B68 C.AAN: .ASCII <9>\any_cylinder\<0><0><0>
00B76
00B78 C.AAO: .ASCII \cluster\<0>
00B80 C.AAP: .ASCII \cylinder\
00B88 C.AAQ: .ASCII \file_ID\<0>
00B90 C.AAR: .ASCII \file_name\<0><0><0>
00B9C C.AAS: .ASCII \logical\<0>
00BA4 C.AAT: .ASCII \virtual\<0>
00BAC C.AAU: .ASCII <9>\none\<0><0><0>
00BB4 C.AAV: .ASCII <9>\indexed\
00BBC C.AAW: .ASCII <9>\relative\<0><0><0>
00BC8 C.AAX: .ASCII <9>\sequential\<0>
00BD4 C.AAY: .ASCII <9>\if_in_recovery_unit\
00BE2
00BE8 C.AAZ: .ASCII <9>\necessary_to_write\<0>
00BF6
00BFC C.ABA: .ASCII <9>\never_RU_journal\<0><0><0>
00C0A
00C10 C.ABB: .ASCII <9>\carriage_return\
00C1E
00C20 C.ABC: .ASCII <9>\FORTRAN\
00C28 C.ABD: .ASCII <9>\print\<0><0>
00C30 C.ABE: .ASCII <9>\fixed\<0><0>
00C38 C.ABF: .ASCII <9>\stream\<0>
00C40 C.ABG: .ASCII <9>\stream_CR\<0><0>
00C4C C.ABH: .ASCII <9>\stream_LF\<0><0>
00C58 C.ABI: .ASCII <9>\undefined\<0><0>
00C64 C.ABJ: .ASCII <9>\variable\<0><0><0>
00C70 C.ABK: .ASCII <9>\VFC\
00C74 C.ABL: .ASCII <9>\bin2\<0><0><0>
00C7C C.ABM: .ASCII <9>\bin4\<0><0><0>
00C84 C.ABN: .ASCII <9>\bin8\<0><0><0>
00C8C C.ABO: .ASCII <9>\decimal\
00C94 C.ABP: .ASCII <9>\int2\<0><0><0>
00C9C C.ABQ: .ASCII <9>\int4\<0><0><0>
00CA4 C.ABR: .ASCII <9>\int8\<0><0><0>
00CAC C.ABS: .ASCII <9>\string\<0>
00CB4 C.ABT: .ASCII <9>\IAS\
00CB8 C.ABU: .ASCII <9>\RSTS/E\<0>
00CC0 C.ABV: .ASCII <9>\RSX-11M\
00CC8 C.ABW: .ASCII <9>\RSX-11M-PLUS\<0><0><0>
00CD6
00CD8 C.ABX: .ASCII <9>\RT-11\<0><0>
00CE0 C.ABY: .ASCII <9>\TRAX-11\
00CE8 C.ABZ: .ASCII <9>\VAX/VMS\
00CF0 C.ACA: .ASCII <9>\yes\
00CF4 C.ACB: .ASCII <9>\no\<0>
00CF8 C.ACC: .ASCII <9>\'\<0><0>
00CFC C.ACD: .LONG ^XA6,0,0,0,0,0,0,0
00D10
00D1C C.ACE: .ASCII <9>\(\<0><0>
00D20 C.ACF: .ASCII \! !X\<0><0><0>
00D28 C.ACG: .ASCII <9>\(system:\<0><0><0>
00D34 C.ACH: .ASCII \, owner:\
00D3C C.ACI: .ASCII \, group:\
00D44 C.ACJ: .ASCII \, world:\

```


Generated Code									
	56	04	AC	D0	000C3	MOVL	4(R12),R6		
	56	04	A6	D0	000C7	MOVL	4(R6),R6		
FFEO	04 B445	FF	A645	90	000CB	MOVB	-1(R6)[INDEX],@4(R4)[INDEX]		
	8F		53	F1	000D2	ACBL	R3,#-1,R1,16\$		
	00FE	04	BC	B1	000DC	17\$:	CMPL	@4(R12),#254	: 0223
			00V	1E	000E2	BGEQU	20\$		
		04	BC	B6	000E4	INCL	@4(R12)	: 0225	
			52	D7	000E7	20\$:	DECL	SCAN_INDEX	: 0229
	01		52	D1	000E9	CMPL	SCAN_INDEX,#1		
			B3	18	000EC	BGEQ	14\$		
	52		50	90	000EE	MOVB	QUOTES,QCHAR	: 0233	
			00V	11	000F1	BRB	25\$		
	52 00000000G		EF	90	000F3	22\$:	MOVB	APOSTROPHE,QCHAR	: 0241
			00V	11	000FA	BRB	25\$		
	52		50	90	000FC	24\$:	MOVB	QUOTES,QCHAR	: 0251
	50	04	BC	3C	000FF	25\$:	MOVZWL	@4(R12),R0	: 0258
			00V	15	00103	BLEQ	27\$		
	55		50	D0	00105	26\$:	MOVL	R0,INDEX	
	51	04	AC	D0	00108	MOVL	4(R12),R1	: 0260	
	53	04	AC	D0	0010C	MOVL	4(R12),R3		
	53	04	A3	D0	00110	MOVL	4(R3),R3		
	04 B145	FF	A345	90	00114	MOVB	-1(R3)[INDEX],@4(R1)[INDEX]		
	E7		50	F5	0011B	SOBGR	R0,26\$		
	04 BC		02	A0	0011E	27\$:	ADDW2	#2,@4(R12)	: 0265
	50	04	AC	D0	00122	MOVL	4(R12),R0	: 0266	
	04 B0		52	90	00126	MOVB	QCHAR,@4(R0)		
	50	04	BC	3C	0012A	MOVZWL	@4(R12),R0	: 0267	
	55	04	AC	D0	0012E	MOVL	4(R12),R5		
	55	04	A5	D0	00132	MOVL	4(R5),R5		
	FF A540		52	90	00136	MOVB	QCHAR,-1(R5)[R0]		
			00V	11	0013B	BRB	30\$		
	FB AD 010E0002		BF	D0	0013D	29\$:	MOVL	#17694722,-8(FP)	: 0277
	FC AD 00000000G		EF	9E	00145	MOVAB	EMPTY_STRING,-4(FP)		
			F8	AD	0014D	PUSHAB	-8(FP)		
			04	AC	DD	00150	PUSHL	4(R12)	
	00000000G		EF	02	FB	00153	CALLS	#2,STRSTRIM	
			04	0015A	30\$:	RET		: 0281	

; Routine Size: 347 bytes. Routine Base: \$CODE + 00D4C

				00000	SHOW_PRIMARY:		: 0328		
			0004	00000	.WORD	*M<R2>			
	5C	00000000G	EF	D0	00002	MOVL	DEF_CURRENT,R12	: 0335	
	52	19	AC	90	00009	MOVB	25(R12),TEMP_PRI	: 0342	
	50		52	9A	0000D	MOVZBL	TEMP_PRI,R0	: 0347	
	7E	00000000GEF	40	9A	00010	MOVZBL	PRIMARY_WIDTH[R0],-(SP)		
	7E		52	9A	00018	MOVZBL	TEMP_PRI,-(SP)		
		FFFFFF138	EF	9F	0001B	PUSHAB	C.AAX		
		00000000G	EF	9F	00021	PUSHAB	FDL_DEST		
	00000000G		EF	04	FB	00027	CALLS	#4,PASSWRITE_ENUMERATED	
	50		52	9A	0002E	MOVZBL	TEMP_PRI,R0	: 0352	
	10		50	D1	00031	CMPL	R0,#T6		
			00V	1E	00034	BGEQU	3\$		
	00VFFFFFF1F3		EF	50	E1	00036	BBC	R0,C.AAB,3\$	
				01	DD	0003E	PUSHL	#1	: 0354
				20	DD	00040	PUSHL	#32	
		00000000G	EF	9F	00042	PUSHAB	FDL_DEST		

Generated Code								
00000000G	EF		03	FB	00048	CALLS	#3,PASSWRITE_CHAR	
		1A	AC	9F	0004F	PUSHAB	26(R12)	
00000000G	EF		01	FB	00052	CALLS	#1,NUM_LEN	
			50	DD	00059	PUSHL	R0	
		1A	AC	DD	0005B	PUSHL	26(R12)	
00000000G	EF	00000000G	EF	9F	0005E	PUSHAB	FDL_DEST	
	52		03	FB	00064	CALLS	#3,PASSWRITE_INTEGER	
	10		52	9A	0006B	MOVZBL	TEMP_PRI,R2	: 0359
00VFFFFFF1B8	EF		00V	1E	00071	CMPL	R2,#T6	
			52	E1	00073	BGEQU	7\$	
0D4C	CF		11	AC	9F	BBC	R2,C.AAC,7\$: 0363
			11	AC	9F	PUSHAB	17(R12)	
				01	FB	CALLS	#1,CHECK_QUOTES	: 0365
				AC	B5	TSTW	17(R12)	: 0367
				00V	15	BLEQ	7\$	
				01	DD	PUSHL	#1	
				09	DD	PUSHL	#9	
00000000G	EF	00000000G	EF	9F	0008C	PUSHAB	FDL_DEST	
	7E		03	FB	00092	CALLS	#3,PASSWRITE_CHAR	
			11	AC	3C	MOVZWL	17(R12),-(SP)	
			00	DD	0009D	PUSHL	#0	
			15	BC	9F	PUSHAB	21(R12)	
		000000FF	8F	DD	000A2	PUSHL	#255	
		00000000G	EF	9F	000AB	PUSHAB	FDL_DEST	
00000000G	EF		05	FB	000AE	CALLS	#5,PASSWRITE_STRING	
			04	000B5	7\$:	RET		: 0374

: Routine Size: 182 bytes, Routine Base: \$CODE + 00EA7

			0000	0000	SHOW_SECONDARY:			: 0421
			000C	0000	.WORD	^M<R2,R3>		
	5C	00000000G	EF	DD	00002	MOVL	DEF_CURRENT,R12	: 0428
	52		1E	AC	90	MOVB	30(R12),TEMP_SEC	: 0435
	50			52	9A	MOVZBL	TEMP_SEC,R0	: 0440
00000098	8F			50	D1	CMPL	R0,#T52	
				00V	1E	BGEQU	2\$	
03 FFFFF15E	EF		50	E1	00019	BBC	R0,C.AAD,..+3	
			0000V	31	00021	BRW	12\$	
				01	DD	2\$:	PUSHL	#1
				09	DD	PUSHL	#9	: 0444
		00000000G	EF	9F	00028	PUSHAB	FDL_DEST	
00000000G	EF		03	FB	0002E	CALLS	#3,PASSWRITE_CHAR	
	50		52	9A	00035	MOVZBL	TEMP_SEC,R0	
	53	00000000G	EF	40	9E	MOVAB	SECONDARY_WIDTH[R0],R3	
	7E		63	9A	00040	MOVZBL	(R3),-(SP)	
	7E		52	9A	00043	MOVZBL	TEMP_SEC,-(SP)	
		FFrFF147	EF	9F	00046	PUSHAB	C.AAE	
		00000000G	EF	9F	0004C	PUSHAB	FDL_DEST	
00000000G	EF		04	FB	00052	CALLS	#4,PASSWRITE_ENUMERATED	
	21		52	91	00059	CMPB	TEMP_SEC,#33	: 0449
			00V	12	0005C	BNEQ	5\$	
00000100	8F	23	AC	D1	0005E	CMPL	35(R12),#256	
			00V	1E	00066	BGEQU	4\$	
00VFFFFFFB4E	EF	23	AC	E0	00068	BBS	35(R12),C.AAF,5\$	
		FFFFFFB68	EF	9F	00071	4\$:	PUSHAB	C.AAG
			04	DD	00077	PUSHL	#4	: 0455
		00000000G	EF	9F	00079	PUSHAB	FDL_DEST	

Generated Code							
00000000G	EF	03	FB	0007F	CALLS	#3,PASSWRITE_STRING	
		0000V	31	00086	BRW	19\$	
	08	63	91	00089	58:	CMPB	(R3),#8 ; 0457
		00V	1E	0008C	BGEQU	7\$	
		FFFFFB4F	FF	0008E	PUSHAB	C.AAH	; 0459
			02	00094	PUSHL	#2	
00000000G	EF	03	FB	00096	PUSHAB	FDL_DEST	
		0000V	31	000A3	CALLS	#3,PASSWRITE_STRING	
	10	63	91	000A6	78:	BRW	19\$; 0461
		03	1F	000A9	CMPB	(R3),#16	
		0000V	31	000AB	BLSSU	.+3	
			01	000AE	BRW	19\$; 0463
			09	000B0	PUSHL	#1	
00000000G	EF	03	FB	000B2	PUSHL	#9	
		0000V	31	000B8	PUSHAB	FDL_DEST	
	85	8F	52	000BF	128:	CALLS	#3,PASSWRITE_CHAR ; 0474
			91	000C2	BRW	19\$	
			12	000C6	CMPB	TEMP_SEC,#-123	
		FFFFFB19	EF	000C8	BNEQ	14\$; 0476
			04	000CE	PUSHAB	C.AAI	
00000000G	EF	03	FB	000D0	PUSHL	#4	
			9F	000D6	PUSHAB	FDL_DEST	
			01	000DD	CALLS	#3,PASSWRITE_STRING	
		1F	AC	000DF	PUSHL	#1	
00000000G	EF	03	FB	000E2	PUSHL	31(R12)	
			9F	000E8	PUSHAB	FDL_DEST	
		FFFFFAF6	EF	000EF	CALLS	#3,PASSWRITE_INTEGER	
			9F	000F5	PUSHAB	C.AAJ	
00000000G	EF	03	FB	000F7	PUSHL	#8	
			9F	000FD	PUSHAB	FDL_DEST	
	86	8F	52	00104	148:	CALLS	#3,PASSWRITE_STRING ; 0481
			91	00108	CMPB	TEMP_SEC,#-122	
			12	0010A	BNEQ	16\$	
		FFFFFAE3	EF	0010A	PUSHAB	C.AAK	; 0483
			04	00110	PUSHL	#4	
00000000G	EF	03	FB	00112	PUSHAB	FDL_DEST	
			9F	00118	CALLS	#3,PASSWRITE_STRING	
			01	0011F	PUSHL	#1	
		1F	AC	00121	PUSHL	31(R12)	
00000000G	EF	03	FB	00124	PUSHAB	FDL_DEST	
			9F	0012A	CALLS	#3,PASSWRITE_INTEGER	
		FFFFFAC0	EF	00131	PUSHAB	C.AAL	
			0A	00137	PUSHL	#10	
00000000G	EF	03	FB	00139	PUSHAB	FDL_DEST	
			9F	0013F	CALLS	#3,PASSWRITE_STRING	
	87	8F	52	00146	168:	CMPB	TEMP_SEC,#-121 ; 0488
			91	0014A	BNEQ	19\$	
			12	0014A	PUSHAB	C.AAM	; 0490
		FFFFFAB1	EF	0014C	PUSHL	#7	
			07	00152	PUSHAB	FDL_DEST	
00000000G	EF	03	FB	00154	CALLS	#3,PASSWRITE_STRING	
			9F	0015A	198:	RET	; 0502
			04	00161			

; Routine Size: 354 bytes, Routine Base: \$CODE + 00F5D

0000 0000 SHOW_QUALIFIER: ; 0548
0000 0000 .WORD ^M<>

```

28      50 00000000G EF DO 00002      MOVL DEF CURRENT,R0      ; 0555
      00      23 AO CF 00009      CASEL 35(R0),#0,#40
      0000V      0000E      .DISPL 1$
      0000V      00010      .DISPL 2$
      0000V      00012      .DISPL 3$
      0000V      00014      .DISPL 4$
      0000V      00016      .DISPL 5$
      0000V      00018      .DISPL 6$
      0000V      0001A      .DISPL 8$
      0000V      0001C      .DISPL 7$
      0000V      0001E      .DISPL 8$
      0000V      00020      .DISPL 15$
      0000V      00022      .DISPL 16$
      0000V      00024      .DISPL 17$
      0000V      00026      .DISPL 22$
      0000V      00028      .DISPL 18$
      0000V      0002A      .DISPL 23$
      0000V      0002C      .DISPL 24$
      0000V      0002E      .DISPL 19$
      0000V      00030      .DISPL 21$
      0000V      00032      .DISPL 20$
      0000V      00034      .DISPL 12$
      0000V      00036      .DISPL 13$
      0000V      00038      .DISPL 14$
      0000V      0003A      .DISPL 33$
      0000V      0003C      .DISPL 34$
      0000V      0003E      .DISPL 35$
      0000V      00040      .DISPL 36$
      0000V      00042      .DISPL 37$
      0000V      00044      .DISPL 38$
      0000V      00046      .DISPL 39$
      0000V      00048      .DISPL 11$
      0000V      0004A      .DISPL 10$
      0000V      0004C      .DISPL 9$
      0052      0004E      .DISPL 82
      0000V      00050      .DISPL 32$
      0000V      00052      .DISPL 29$
      0000V      00054      .DISPL 25$
      0000V      00056      .DISPL 30$
      0000V      00058      .DISPL 26$
      0000V      0005A      .DISPL 31$
      0000V      0005C      .DISPL 27$
      0000V      0005E      .DISPL 28$
      0000V      31 00060      BRW      40$
      FFFFA40 EF 9F 00063 1$: PUSHAB C.AAN      ; 0557
      00000000G EF DD 00069      PUSHL #13
      00000000G EF 9F 0006B      PUSHAB FDL_DEST
      0000V      03 FB 00071      CALLS #3,PASSWRITE_STRING
      FFFFA38 EF 9F 0007B 2$: BRW      41$      ; 0558
      00000000G EF DD 00081      PUSHAB C.AAO
      0000V      07 DD 00081      PUSHL #7
      00000000G EF 9F 00083      PUSHAB FDL_DEST
      0000V      03 FB 00089      CALLS #3,PASSWRITE_STRING
      FFFFA28 EF 9F 00093 3$: BRW      41$      ; 0559
      00000000G EF DD 00099      PUSHAB C.AAP
      0000V      08 DD 00099      PUSHL #8
      00000000G EF 9F 0009B      PUSHAB FDL_DEST

```

Generated Code						
00000000G	EF	03	FB 000A1	CALLS	#3,PASSWRITE_STRING	
		0000V	31 000A8	BRW	41\$	
	FFFFFFA18	EF	9F 000AB	4\$: PUSHAB	C.AAQ	: 0560
		07	DD 000B1	PUSHL	#7	
00000000G	EF	03	FB 000B3	PUSHAB	FDL_DEST	
		0000V	31 000B9	CALLS	#3,PASSWRITE_STRING	
	FFFFFFA08	EF	9F 000C0	BRW	41\$	
		09	DD 000C3	5\$: PUSHAB	C.AAR	: 0561
		07	DD 000C9	PUSHL	#9	
00000000G	EF	03	FB 000CB	PUSHAB	FDL_DEST	
		0000V	31 000D1	CALLS	#3,PASSWRITE_STRING	
	FFFFFF9FC	EF	9F 000D8	BRW	41\$	
		07	DD 000DB	6\$: PUSHAB	C.AAS	: 0562
		07	DD 000E1	PUSHL	#7	
00000000G	EF	03	FB 000E3	PUSHAB	FDL_DEST	
		0000V	31 000E9	CALLS	#3,PASSWRITE_STRING	
	FFFFFF9EC	EF	9F 000F0	BRW	41\$	
		07	DD 000F3	7\$: PUSHAB	C.AAT	: 0563
		07	DD 000F9	PUSHL	#7	
00000000G	EF	03	FB 000FB	PUSHAB	FDL_DEST	
		0000V	31 00101	CALLS	#3,PASSWRITE_STRING	
	FFFFFF9DC	EF	9F 00108	BRW	41\$	
		05	DD 0010B	8\$: PUSHAB	C.AAU	: 0565
		05	DD 00111	PUSHL	#5	
00000000G	EF	03	FB 00113	PUSHAB	FDL_DEST	
		0000V	31 00119	CALLS	#3,PASSWRITE_STRING	
	FFFFFF9CC	EF	9F 00120	BRW	41\$	
		08	DD 00123	9\$: PUSHAB	C.AAV	: 0566
		08	DD 00129	PUSHL	#8	
00000000G	EF	03	FB 0012B	PUSHAB	FDL_DEST	
		0000V	31 00131	CALLS	#3,PASSWRITE_STRING	
	FFFFFF9BC	EF	9F 00138	BRW	41\$	
		09	DD 0013B	10\$: PUSHAB	C.AAW	: 0567
		09	DD 00141	PUSHL	#9	
00000000G	EF	03	FB 00143	PUSHAB	FDL_DEST	
		0000V	31 00149	CALLS	#3,PASSWRITE_STRING	
	FFFFFF9B0	EF	9F 00150	BRW	41\$	
		08	DD 00153	11\$: PUSHAB	C.AAX	: 0568
		08	DD 00159	PUSHL	#11	
00000000G	EF	03	FB 0015B	PUSHAB	FDL_DEST	
		0000V	31 00161	CALLS	#3,PASSWRITE_STRING	
	FFFFFF9A4	EF	9F 00168	BRW	41\$	
		14	DD 0016B	12\$: PUSHAB	C.AAY	: 0569
		14	DD 00171	PUSHL	#20	
00000000G	EF	03	FB 00173	PUSHAB	FDL_DEST	
		0000V	31 00179	CALLS	#3,PASSWRITE_STRING	
	FFFFFF9A0	EF	9F 00180	BRW	41\$	
		13	DD 00183	13\$: PUSHAB	C.AAZ	: 0570
		13	DD 00189	PUSHL	#19	
00000000G	EF	03	FB 0018B	PUSHAB	FDL_DEST	
		0000V	31 00191	CALLS	#3,PASSWRITE_STRING	
	FFFFFF99C	EF	9F 00198	BRW	41\$	
		11	DD 0019B	14\$: PUSHAB	C.ABA	: 0571
		11	DD 001A1	PUSHL	#17	
00000000G	EF	03	FB 001A3	PUSHAB	FDL_DEST	
		0000V	31 001A9	CALLS	#3,PASSWRITE_STRING	
		0000V	31 001B0	BRW	41\$	

Generated Code								
	FFFFFF998	EF	9F	001B3	15\$:	PUSHAB	C.ABB	: 0572
		10	DD	001B9		PUSHL	#16	
0000000G	0000000G	EF	9F	001BB		PUSHAB	FDL_DEST	
		03	FB	001C1		CALLS	#3,PASSWRITE_STRING	
		0000V	31	001C8		BRW	41\$	
	FFFFFF990	EF	9F	001CB	16\$:	PUSHAB	C.ABC	: 0573
		08	DD	001D1		PUSHL	#8	
0000000G	0000000G	EF	9F	001D3		PUSHAB	FDL_DEST	
		03	FB	001D9		CALLS	#3,PASSWRITE_STRING	
		0000V	31	001E0		BRW	41\$	
	FFFFFF980	EF	9F	001E3	17\$:	PUSHAB	C.ABD	: 0574
		06	DD	001E9		PUSHL	#6	
0000000G	0000000G	EF	9F	001EB		PUSHAB	FDL_DEST	
		03	FB	001F1		CALLS	#3,PASSWRITE_STRING	
		0000V	31	001F8		BRW	41\$	
	FFFFFF970	EF	9F	001FB	18\$:	PUSHAB	C.ABE	: 0575
		06	DD	00201		PUSHL	#6	
0000000G	0000000G	EF	9F	00203		PUSHAB	FDL_DEST	
		03	FB	00209		CALLS	#3,PASSWRITE_STRING	
		0000V	31	00210		BRW	41\$	
	FFFFFF960	EF	9F	00213	19\$:	PUSHAB	C.ABF	: 0576
		07	DD	00219		PUSHL	#7	
0000000G	0000000G	EF	9F	0021B		PUSHAB	FDL_DEST	
		03	FB	00221		CALLS	#3,PASSWRITE_STRING	
		0000V	31	00228		BRW	41\$	
	FFFFFF950	EF	9F	0022B	20\$:	PUSHAB	C.ABG	: 0577
		0A	DD	00231		PUSHL	#10	
0000000G	0000000G	EF	9F	00233		PUSHAB	FDL_DEST	
		03	FB	00239		CALLS	#3,PASSWRITE_STRING	
		0000V	31	00240		BRW	41\$	
	FFFFFF944	EF	9F	00243	21\$:	PUSHAB	C.ABH	: 0578
		0A	DD	00249		PUSHL	#10	
0000000G	0000000G	EF	9F	0024B		PUSHAB	FDL_DEST	
		03	FB	00251		CALLS	#3,PASSWRITE_STRING	
		0000V	31	00258		BRW	41\$	
	FFFFFF938	EF	9F	0025B	22\$:	PUSHAB	C.ABI	: 0579
		0A	DD	00261		PUSHL	#10	
0000000G	0000000G	EF	9F	00263		PUSHAB	FDL_DEST	
		03	FB	00269		CALLS	#3,PASSWRITE_STRING	
		0000V	31	00270		BRW	41\$	
	FFFFFF92C	EF	9F	00273	23\$:	PUSHAB	C.ABJ	: 0580
		09	DD	00279		PUSHL	#9	
0000000G	0000000G	EF	9F	0027B		PUSHAB	FDL_DEST	
		03	FB	00281		CALLS	#3,PASSWRITE_STRING	
		0000V	31	00288		BRW	41\$	
	FFFFFF920	EF	9F	0028B	24\$:	PUSHAB	C.ABK	: 0581
		04	DD	00291		PUSHL	#4	
0000000G	0000000G	EF	9F	00293		PUSHAB	FDL_DEST	
		03	FB	00299		CALLS	#3,PASSWRITE_STRING	
		0000V	31	002A0		BRW	41\$	
	FFFFFF90C	EF	9F	002A3	25\$:	PUSHAB	C.ABL	: 0582
		05	DD	002A9		PUSHL	#5	
0000000G	0000000G	EF	9F	002AB		PUSHAB	FDL_DEST	
		03	FB	002B1		CALLS	#3,PASSWRITE_STRING	
		0000V	31	002B8		BRW	41\$	
	FFFFFF8FC	EF	9F	002BB	26\$:	PUSHAB	C.ABM	: 0583
		05	DD	002C1		PUSHL	#5	

Generated Code					
0000000G	EF	0000000G	EF	9F 002C3	PUSHAB FDL DEST
			03	FB 002C9	CALLS #3,PASSWRITE_STRING
			0000V	31 002D0	BRW 41\$
		FFFFF8EC	EF	9F 002D3	27\$: PUSHAB C.ABN ; 0584
			05	DD 002D9	PUSHL #5
0000000G	EF	0000000G	EF	9F 002DB	PUSHAB FDL DEST
			03	FB 002E1	CALLS #3,PASSWRITE_STRING
			0000V	31 002E8	BRW 41\$
		FFFFF8DC	EF	9F 002EB	28\$: PUSHAB C.ABO ; 0585
			08	DD 002F1	PUSHL #8
0000000G	EF	0000000G	EF	9F 002F3	PUSHAB FDL DEST
			03	FB 002F9	CALLS #3,PASSWRITE_STRING
			0000V	31 00300	BRW 41\$
		FFFFF8CC	EF	9F 00303	29\$: PUSHAB C.ABP ; 0586
			05	DD 00309	PUSHL #5
0000000G	EF	0000000G	EF	9F 0030B	PUSHAB FDL DEST
			03	FB 00311	CALLS #3,PASSWRITE_STRING
			0000V	31 00318	BRW 41\$
		FFFFF8BC	EF	9F 0031B	30\$: PUSHAB C.ABQ ; 0587
			05	DD 00321	PUSHL #5
0000000G	EF	0000000G	EF	9F 00323	PUSHAB FDL DEST
			03	FB 00329	CALLS #3,PASSWRITE_STRING
			0000V	31 00330	BRW 41\$
		FFFFF8AC	EF	9F 00333	31\$: PUSHAB C.ABR ; 0588
			05	DD 00339	PUSHL #5
0000000G	EF	0000000G	EF	9F 0033B	PUSHAB FDL DEST
			03	FB 00341	CALLS #3,PASSWRITE_STRING
			0000V	31 00348	BRW 41\$
		FFFFF89C	EF	9F 0034B	32\$: PUSHAB C.ABS ; 0589
			07	DD 00351	PUSHL #7
0000000G	EF	0000000G	EF	9F 00353	PUSHAB FDL DEST
			03	FB 00359	CALLS #3,PASSWRITE_STRING
			0000V	31 00360	BRW 41\$
		FFFFF88C	EF	9F 00363	33\$: PUSHAB C.ABT ; 0590
			04	DD 00369	PUSHL #4
0000000G	EF	0000000G	EF	9F 0036B	PUSHAB FDL DEST
			03	FB 00371	CALLS #3,PASSWRITE_STRING
			0000V	31 00378	BRW 41\$
		FFFFF878	EF	9F 0037B	34\$: PUSHAB C.ABU ; 0591
			07	DD 00381	PUSHL #7
0000000G	EF	0000000G	EF	9F 00383	PUSHAB FDL DEST
			03	FB 00389	CALLS #3,PASSWRITE_STRING
			00V	11 00390	BRB 41\$
		FFFFF869	EF	9F 00392	35\$: PUSHAB C.ABV ; 0592
			08	DD 00398	PUSHL #8
0000000G	EF	0000000G	EF	9F 0039A	PUSHAB FDL DEST
			03	FB 003A0	CALLS #3,PASSWRITE_STRING
			00V	11 003A7	BRB 41\$
		FFFFF85A	EF	9F 003A9	36\$: PUSHAB C.ABW ; 0593
			0D	DD 003AF	PUSHL #13
0000000G	EF	0000000G	EF	9F 003B1	PUSHAB FDL DEST
			03	FB 003B7	CALLS #3,PASSWRITE_STRING
			00V	11 003BE	BRB 41\$
		FFFFF853	EF	9F 003C0	37\$: PUSHAB C.ABX ; 0594
			06	DD 003C6	PUSHL #6
0000000G	EF	0000000G	EF	9F 003C8	PUSHAB FDL DEST
			03	FB 003CE	CALLS #3,PASSWRITE_STRING

Generated Code

			00V	11	003D5		BRB	41\$		
		FFFFF844	EF	9F	003D7	38\$:	PUSHAB	C.ABY		: 0535
			08	DD	003DD		PUSHL	#8		
00000000G	EF	00000000G	EF	9F	003DF		PUSHAB	FDL_DEST		
			03	FB	003E5		CALLS	#3,PASSWRITE_STRING		
		FFFFF835	00V	11	003EC		BRB	41\$		
			EF	9F	003EE	39\$:	PUSHAB	C.ABZ		: 0596
			08	DD	003F4		PUSHL	#8		
00000000G	EF	00000000G	EF	9F	003F6		PUSHAB	FDL_DEST		
			03	FB	003FC		CALLS	#3,PASSWRITE_STRING		
			00V	11	00403		BRB	41\$		
					00405	40\$:				
					00405	41\$:	RET			: 0604
; Routine Size: 1030 bytes, Routine Base: \$CODE + 010BF										
					00000		SHOW_CURRENT:			: 0656
				0004	00000		.WORD	^M<R2>		
	5E			14	C2	00002	SUBL2	#20,SP		
	5C	04		BC	90	00005	MOVB	@4(R12),PLUS_VALUE		
00V00000000G	EF			00	E1	00009	BBC	#0,DEST_IS_TERMINAL,2\$: 0688
		00000000G	EF	9F	00011		PUSHAB	SHIFT		: 0690
				04	DD	00017	PUSHL	#4		
		00000000G	EF	9F	00019		PUSHAB	FDL_DEST		
00000000G	EF			03	FB	0001F	CALLS	#3,PASSWRITE_STRING		
	50	00000000G	EF	D0	00026	2\$:	MOVL	DEF_CURRENT,R0		: 0695
	50			60	9A	0002D	MOVZBL	(R0),R0		
02	00			50	8F	00030	CASEB	R0,#0,#2		
				0000V		00034	.DISPL	3\$		
				0000V		00036	.DISPL	9\$		
				0000V		00038	.DISPL	51\$		
				0000V	31	0003A	BRW	55\$		
	52	00000000G	EF	D0	0003D	3\$:	MOVL	DEF_CURRENT,R2		: 0699
0EA7	CF			00	FB	00044	CALLS	#0,SHOW_PRIMARY		: 0706
	00V			5C	E9	00049	BLBC	PLUS_VALUE,8\$: 0711
		09		A2	B5	0004C	TSTW	9(R2)		: 0716
				00V	15	0004F	BLEQ	8\$		
				01	DD	00051	PUSHL	#1		: 0718
				09	DD	00053	PUSHL	#9		
		00000000G	EF	9F	00055		PUSHAB	FDL_DEST		
00000000G	EF			03	FB	0005B	CALLS	#3,PASSWRITE_CHAR		
	7E	09		A2	3C	00062	MOVZWL	9(R2),-(SP)		
				00	DD	00066	PUSHL	#0		
		0D		B2	9F	00068	PUSHAB	@13(R2)		
		000000FF		8F	DD	0006B	PUSHL	#255		
		00000000G	EF	9F	00071		PUSHAB	FDL_DEST		
00000000G	EF			05	FB	00077	CALLS	#5,PASSWRITE_STRING		
		00000000G	EF	9F	0007E	8\$:	PUSHAB	FDL_DEST		: 0724
00000000G	EF			01	FB	00084	CALLS	#1,PASSWRITELN2		
				0000V	31	0008B	BRW	56\$		
	52	00000000G	EF	D0	0008E	9\$:	MOVL	DEF_CURRENT,R2		: 0730
0F5D	CF			00	FB	00095	CALLS	#0,SHOW_SECONDARY		: 0737
	03			5C	E8	0009A	BLBS	PLUS_VALUE,..+3		: 0742
				0000V	31	0009D	BRW	50\$		
	5C	1E		A2	9A	000A0	MOVZBL	30(R2),R12		: 0749
	5C			04	C4	000A4	MULL2	#4,R12		
00V00000000G	EF			5C	E1	000A7	BBC	R12,SEC_TYPE,15\$		

Generated Code										
OD4C	CF	11	A2	9F	000AF	PUSHAB	17(R2)			: 0753
			01	FB	000B2	CALLS	#1,CHECK_QUOTES			
		11	A2	B5	000B7	TSTW	17(R2)			: 0755
			00V	15	000BA	BLEQ	15\$			
			01	DD	000BC	PUSHL	#1			: 0757
			09	DD	000BE	PUSHL	#9			
00000000G	EF	00000000G	EF	9F	000C0	PUSHAB	FDL_DEST			
	7E		03	FB	000C6	CALLS	#3,PASSWRITE_CHAR			
			11	A2	3C	000CD	MOVZWL	17(R2),-(SP)		
			00	DD	000D1	PUSHL	#0			
			15	B2	9F	000D3	PUSHAB	@21(R2)		
		000000FF	8F	DD	000D6	PUSHL	#255			
00000000G	EF	00000000G	EF	9F	000DC	PUSHAB	FDL_DEST			
			05	FB	000E2	CALLS	#5,PASSWRITE_STRING			
	50		1E	A2	9A	000E9	MOVZBL	30(R2),R0		: 0765
	50			04	C4	000ED	MULL2	#4,R0		
	50			02	C0	000F0	ADDL2	#2,R0		
00V00000000G	EF		50	E1	000F3	BBC	R0,SEC_TYPE,17\$			
10BF	CF		00	FB	000FB	CALLS	#0,SHOW_QUALIFIER			: 0767
	50		1E	A2	9A	00100	MOVZBL	30(R2),R0		: 0772
	50			04	C4	00104	MULL2	#4,R0		
			50	D6	00107	INCL	R0			
00V00000000G	EF		50	E1	00109	BBC	R0,SEC_TYPE,19\$			
			01	DD	00111	PUSHL	#1			: 0777
			09	DD	00113	PUSHL	#9			
00000000G	EF	00000000G	EF	9F	00115	PUSHAB	FDL_DEST			
			03	FB	0011B	CALLS	#3,PASSWRITE_CHAR			
00000000G	EF		27	A2	9F	00122	PUSHAB	39(R2)		
				01	FB	00125	CALLS	#1,NUM_LEN		
			50	DD	0012C	PUSHL	R0			
			27	A2	DD	0012E	PUSHL	39(R2)		
00000000G	EF	00000000G	EF	9F	00131	PUSHAB	FDL_DEST			
			03	FB	00137	CALLS	#3,PASSWRITE_INTEGER			
	50		1E	A2	9A	0013E	MOVZBL	30(R2),R0		: 0782
	50			04	C4	00142	MULL2	#4,R0		
	50			03	C0	00145	ADDL2	#3,R0		
00V00000000G	EF		50	E1	00148	BBC	R0,SEC_TYPE,24\$			
00V	2B	A2	00	E1	00150	BBC	#0,43(R2),22\$: 0784
			FFFFF6D0	EF	9F	00155	PUSHAB	C.ACA		: 0786
				04	DD	0015B	PUSHL	#4		
00000000G	EF	00000000G	EF	9F	0015D	PUSHAB	FDL_DEST			
			03	FB	00163	CALLS	#3,PASSWRITE_STRING			
			00V	11	0016A	BRB	24\$			
			FFFFF6BD	EF	9F	0016C	PUSHAB	C.ACB		: 0790
				03	DD	00172	PUSHL	#3		
00000000G	EF	00000000G	EF	9F	00174	PUSHAB	FDL_DEST			
			03	FB	0017A	CALLS	#3,PASSWRITE_STRING			
	83		1E	A2	91	00181	CMPB	30(R2),#-125		: 0792
				00V	13	00186	BEQL	26\$		
	59		1E	A2	91	00188	CMPB	30(R2),#89		
				00V	12	0018D	BNEQ	31\$		
			20	A2	D1	0018F	CMPL	39(R2),#32		: 0800
				00V	19	00193	BLSS	28\$		
27	A2	7E	8F	07	00	ED	00195	CMPZV	#0,#7,#*X7E,39(R2)	
					00V	18	0019C	BGEQ	29\$	
					01	DD	0019E	PUSHL	#1	: 0806
					09	DD	001A0	PUSHL	#9	

Generated Code								
00000000G	EF	00000000G	EF	9F	001A2	PUSHAB	FDL_DEST	
		27	03	FB	001A8	CALLS	#3,PASSWRITE_CHAR	
00000000G	EF		A2	9F	001AF	PUSHAB	39(R2)	
			01	FB	001B2	CALLS	#1,NUM_LEN	
		27	50	DD	001B9	PUSHL	R0	
			A2	DD	001BB	PUSHL	39(R2)	
00000000G	EF	00000000G	EF	9F	001BE	PUSHAB	FDL_DEST	
			03	FB	001C4	CALLS	#3,PASSWRITE_INTEGER	
		FFFFF660	00V	11	001CB	BRB	31\$	
			EF	9F	001CD	PUSHAB	C,ACC	: 0810
			02	DD	001D3	PUSHL	#2	
00000000G	EF	00000000G	EF	9F	001D5	PUSHAB	FDL_DEST	
			03	FB	001DB	CALLS	#3,PASSWRITE_STRING	
		27	01	DD	001E2	PUSHL	#1	
			A2	DD	001E4	PUSHL	39(R2)	
00000000G	EF	00000000G	EF	9F	001E7	PUSHAB	FDL_DEST	
			03	FB	001ED	CALLS	#3,PASSWRITE_CHAR	
			01	DD	001F4	PUSHL	#1	
		27	DD	001F6	PUSHL	#39		
00000000G	EF	00000000G	EF	9F	001F8	PUSHAB	FDL_DEST	
			03	FB	001FE	CALLS	#3,PASSWRITE_CHAR	
		21	A2	91	00205	CMPB	30(R2),#33	: 0817
			03	13	00209	BEQL	.+3	
			0000V	31	0020B	BRW	42\$	
10BF	CF		00	FB	0020E	CALLS	#0,SHOW_QUALIFIER	: 0821
00000100	8F	23	A2	D1	00213	CMPB	35(R2),#256	: 0823
			00V	1E	0021B	BGEQU	34\$	
00VFFFFFF611	EF	23	A2	E1	0021D	BBC	35(R2),C.ACD,34\$	
			01	DD	00226	PUSHL	#1	: 0830
			09	DD	00228	PUSHL	#9	
00000000G	EF	00000000G	EF	9F	0022A	PUSHAB	FDL_DEST	
			03	FB	00230	CALLS	#3,PASSWRITE_CHAR	
00000000G	EF	27	A2	9F	00237	PUSHAB	39(R2)	
			01	FB	0023A	CALLS	#1,NUM_LEN	
		27	50	DD	00241	PUSHL	R0	
			A2	DD	00243	PUSHL	39(R2)	
00000000G	EF	00000000G	EF	9F	00246	PUSHAB	FDL_DEST	
			03	FB	0024C	CALLS	#3,PASSWRITE_INTEGER	
			0000V	31	00253	BRW	42\$	
01	03	23	A2	CF	00256	CASEL	35(R2),#3,#1	: 0836
			0000V		0025B	.DISPL	35\$	
			0000V		0025D	.DISPL	36\$	
			0000V	31	0025F	BRW	39\$	
		FFFFF5EF	EF	9F	00262	PUSHAB	C,ACE	: 0842
			02	DD	00268	PUSHL	#2	
00000000G	EF	00000000G	EF	9F	0026A	PUSHAB	FDL_DEST	
			03	FB	00270	CALLS	#3,PASSWRITE_STRING	
00000000G	EF	34	A2	9F	00277	PUSHAB	52(R2)	
			01	FB	0027A	CALLS	#1,NUM_LEN	
		34	50	DD	00281	PUSHL	R0	
			A2	DD	00283	PUSHL	52(R2)	
00000000G	EF	00000000G	EF	9F	00286	PUSHAB	FDL_DEST	
			03	FB	0028C	CALLS	#3,PASSWRITE_INTEGER	
			01	DD	00293	PUSHL	#1	
			2C	DD	00295	PUSHL	#44	
00000000G	EF	00000000G	EF	9F	00297	PUSHAB	FDL_DEST	
			03	FB	0029D	CALLS	#3,PASSWRITE_CHAR	

Generated Code

E 2
16-Sep-1984 01:05:40
5-Sep-1984 13:38:00

VAX-11 Pascal V2.4-277
DISK\$VMSMASTER:[EDF.SRC]EDFSHOW.PAS;1 (22)

		0000V	31	003A7	BRW	47\$	
	FFFFF4B3	EF	9F	003AA	PUSHAB	C.ACG	: 0893
		09	DD	003B0	PUSHL	#9	
00000000G	00000000G	EF	9F	003B2	PUSHAB	FDL_DEST	
		03	FB	003B8	CALLS	#3,PASSWRITE_STRING	
	00000000	8F	DF	003BF	PUSHAL	#0	: 0894
	30	A2	9F	003C5	PUSHAB	48(R2)	
0000V	CF	02	FB	003C8	CALLS	#2,SHOW_PROT	
	FFFFF49C	EF	9F	003CD	PUSHAB	C.ACH	: 0895
		08	DD	003D3	PUSHL	#8	
00000000G	00000000G	EF	9F	003D5	PUSHAB	FDL_DEST	
		03	FB	003DB	CALLS	#3,PASSWRITE_STRING	
	00000004	8F	DF	003E2	PUSHAL	#4	: 0896
	30	A2	9F	003E8	PUSHAB	48(R2)	
0000V	CF	02	FB	003EB	CALLS	#2,SHOW_PROT	
	FFFFF481	EF	9F	003F0	PUSHAB	C.ACI	: 0897
		08	DD	003F6	PUSHL	#8	
00000000G	00000000G	EF	9F	003F8	PUSHAB	FDL_DEST	
		03	FB	003FE	CALLS	#3,PASSWRITE_STRING	
	00000008	8F	DF	00405	PUSHAL	#8	: 0898
	30	A2	9F	0040B	PUSHAB	48(R2)	
0000V	CF	02	FB	0040E	CALLS	#2,SHOW_PROT	
	FFFFF466	EF	9F	00413	PUSHAB	C.ACJ	: 0899
		08	DD	00419	PUSHL	#8	
00000000G	00000000G	EF	9F	0041B	PUSHAB	FDL_DEST	
		03	FB	00421	CALLS	#3,PASSWRITE_STRING	
	0000000C	8F	DF	00428	PUSHAL	#12	: 0900
	30	A2	9F	0042E	PUSHAB	48(R2)	
0000V	CF	02	FB	00431	CALLS	#2,SHOW_PROT	
		01	DD	00436	PUSHL	#1	: 0901
		29	DD	00438	PUSHL	#41	
00000000G	00000000G	EF	9F	0043A	PUSHAB	FDL_DEST	
		03	FB	00440	CALLS	#3,PASSWRITE_CHAR	
	09	A2	B5	00447	TSTW	9(R2)	: 0908
		00V	15	0044A	BLEQ	49\$	
		01	DD	0044C	PUSHL	#1	: 0910
		09	DD	0044E	PUSHL	#9	
00000000G	00000000G	EF	9F	00450	PUSHAB	FDL_DEST	
		03	FB	00456	CALLS	#3,PASSWRITE_CHAR	
	09	A2	3C	0045D	MOVZWL	9(R2),-(SP)	
		00	DD	00461	PUSHL	#0	
	0D	B2	9F	00463	PUSHAB	@13(R2)	
	000000FF	8F	DD	00466	PUSHL	#255	
00000000G	00000000G	EF	9F	0046C	PUSHAB	FDL_DEST	
		05	FB	00472	CALLS	#5,PASSWRITE_STRING	
00000000G	00000000G	EF	9F	00479	PUSHAB	FDL_DEST	: 0916
		01	FB	0047F	CALLS	#1,PASSWRITELN2	
		00V	11	00486	BRB	56\$	
50	00000000G	EF	DD	00488	MOVL	DEF_CURRENT,R0	: 0924
		A0	B5	0048F	TSTW	9(R0)	: 0931
		00V	15	00492	BLEQ	56\$	
	09	A0	3C	00494	MOVZWL	9(R0),-(SP)	: 0933
		00	DD	00498	PUSHL	#0	
	0D	B0	9F	0049A	PUSHAB	@13(R0)	
	000000FF	8F	DD	0049D	PUSHL	#255	
00000000G	00000000G	EF	9F	004A3	PUSHAB	FDL_DEST	
		05	FB	004A9	CALLS	#5,PASSWRITE_STRING	

Generated Code

00000000G	EF	00000000G	EF	9F	004B0		PUSHAB	FDL_DEST	
			01	FB	004B6		CALLS	#1,PASSWRITELN2	
			00V	11	004BD		BRB	56\$	
					004BF	55\$:			
03	00000000G	EF	00	D6	004BF	56\$:	INCL	LINES_SHOWN	: 0947
			0000V	E0	004C5		BBS	#0_DEST_IS_TERMINAL,.,+3	: 0949
				31	004CD		BRW	66\$	
	50	00000000G	EF	D0	004D0		MOVL	DEF_CURRENT,R0	: 0953
			01	A0	004D7		TSTL	1(R0)	
			00V	13	004DA		BEQL	63\$	
50	00000000G	EF	03	C3	004DC		SUBL3	LINES_SHOWN,LINES_PER_PAGE,R0	: 0955
			04	S0	004E8		SUBL2	#3,R0	
				D1	004EB		CMPL	R0,#4	
			00V	18	004EE		BGEQ	63\$	
	50	00000000G	EF	D0	004F0		MOVL	DEF_CURRENT,R0	
			52	D0	004F7		MOVL	DEF_CURRENT,R2	
			52	A2	004FE		MOVL	1(R2),R2	
	19	A2	19	A0	00502		CMPB	25(R0),25(R2)	
			00V	12	00507		BNEQ	61\$	
	52	00000000G	EF	D0	00509		MOVL	DEF_CURRENT,R2	
			50	D0	00510		MOVL	DEF_CURRENT,R0	
			50	A0	00517		MOVL	1(R0),R0	
	1A	A0	1A	A2	0051B		CMPL	26(R2),26(R0)	
			00V	13	00520		BEQL	63\$	
		00000000G	EF	D4	00522	61\$:	CLRL	LINES_SHOWN	: 0969
		00000002	8F	DF	00528		PUSHAL	#2	: 0970
	00000000G	EF	01	FB	0052E		CALLS	#1,CLEAR	
50	00000000G	EF	03	C3	00535	63\$:	SUBL3	#3,LINES_PER_PAGE,R0	: 0974
			50	D1	0053D		CMPL	LINES_SHOWN,R0	
			00V	19	00544		BLSS	66\$	
		00000000G	EF	D4	00546		CLRL	LINES_SHOWN	: 0984
		00000002	8F	DF	0054C		PUSHAL	#2	: 0985
	00000000G	EF	01	FB	00552		CALLS	#1,CLEAR	
			04	00559	66\$:		RET		: 0991

; Routine Size: 1370 bytes, Routine Base: \$CODE + 014C5

				0000	SHOW_PROT:				: 0661
			0004	0000	.WORD				
	52	04	BC	D0	00002		MOVL	@4(R12),PROTECTION	
			5C	D0	00006		MOVL	@8(R12),FIELD_OFFSET	
00V	52	08	5C	E1	0000A		BBC	FIELD_OFFSET,PROTECTION,2\$: 0665
			01	DD	0000E		PUSHL	#1	: 0667
	7E	52	8F	9A	00010		MOVZBL	#82,-(SP)	
		00000000G	EF	9F	00014		PUSHAB	FDL_DEST	
			50	FB	0001A		CALLS	#3,PASSWRITE_CHAR	
			00V	9E	00021	2\$:	MOVAB	1(FIELD_OFFSET),R0	: 0669
			52	E1	00025		BBC	R0,PROTECTION,4\$	
			01	DD	00029		PUSHL	#1	: 0671
	7E	57	8F	9A	0002B		MOVZBL	#87,-(SP)	
		00000000G	EF	9F	0002F		PUSHAB	FDL_DEST	
			50	FB	00035		CALLS	#3,PASSWRITE_CHAR	
			00V	9E	0003C	4\$:	MOVAB	2(FIELD_OFFSET),R0	: 0673
			52	E1	00040		BBC	R0,PROTECTION,6\$	
			01	DD	00044		PUSHL	#1	: 0675
	7E	45	8F	9A	00046		MOVZBL	#69,-(SP)	
		00000000G	EF	9F	0004A		PUSHAB	FDL_DEST	

Generated Code

00000000G	EF		03	FB	00050	CALLS	#3,PASSWRITE_CHAR	
	5C	03	AC	9E	00057	6\$:	MOVAB	3(FIELD_OFFSET),R12 ; 0677
00V	52		5C	E1	0005B		BBC	R12,PROTECTION,8\$
	7E	44	01	DD	0005F		PUSHL	#1 ; 0679
			8F	9A	00061		MOVZBL	#68,-(SP)
00000000G	EF	00000000G	EF	9F	00065		PUSHAB	FDL_DEST
			03	FB	0006B		CALLS	#3,PASSWRITE_CHAR
				04	00072	8\$:	RET	; 0681

: Routine Size: 115 bytes, Routine Base: \$CODE + 01A1F

				00000	GENERATE_FDL:			; 1041
			0004	00000	.WORD		^M<R2>	
00000000G	EF	00000000G	EF	D0	00002		MOVL	DEF_HEAD,DEF_CURRENT ; 1052
	5C		00	D2	0000D		MCOML	#0,PREV_PRINUM ; 1057
		00000000G	EF	D4	00010		CLRL	LINES_SHOWN ; 1062
		00000000G	EF	D5	00016		TSTL	DEF_CURRENT ; 1067
		00V	13	0001C			BEQL	11\$
	50	00000000G	EF	D0	0001E	2\$:	MOVL	DEF_CURRENT,R0 ; 1082
			5C	D5	00025		TSTL	PREV_PRINUM ; 1086
		00V	18	00027			BGEQ	5\$
	52	19	A0	90	00029		MOVB	25(R0),PREV_PRIMARY ; 1093
	5C	1A	A0	D0	0002D		MOVL	26(R0),PREV_PRINUM ; 1094
		00V	11	00031			BRB	9\$
1A	A0		5C	D1	00033	5\$:	CMPL	PREV_PRINUM,26(R0) ; 1105
		00V	12	00037			BNEQ	7\$
19	A0		52	91	00039		CMPB	PREV_PRIMARY,25(R0)
		00V	13	0003D			BEQL	9\$
	52	19	A0	90	0003F	7\$:	MOVB	25(R0),PREV_PRIMARY ; 1111
	5C	1A	A0	D0	00043		MOVL	26(R0),PREV_PRINUM ; 1112
		00000000G	EF	9F	00047		PUSHAB	FDL_DEST ; 1117
00000000G	EF	00000000G	01	FB	0004D		CALLS	#1,PASSWRITELN2
		00000000G	EF	D6	00054		INCL	LINES_SHOWN ; 1118
		01	8F	9F	0005A	9\$:	PUSHAB	#1 ; 1126
14C5	CF		01	FB	0005D		CALLS	#1,SHOW_CURRENT
00000000G	EF	00000000G	00	FB	00062		CALLS	#0,INCR_CURRENT ; 1127
		00000000G	EF	D5	00069		TSTL	DEF_CURRENT
			AD	12	0006F		BNEQ	2\$
				04	00071	11\$:	RET	; 1133

: Routine Size: 114 bytes, Routine Base: \$CODE + 01A92

				00000	VIEW_DEF:			; 1183
			0000	00000	.WORD		^M<>	
00000000G	EF	00000003	8F	DF	00002		PUSHAL	#3 ; 1190
00000000G	EF		01	FB	00008		CALLS	#1,CLEAR
		00V	01	90	0000F		MOVB	#1,DEST_IS_TERMINAL ; 1196
			AF	9F	00016		PUSHAB	1\$; 1198
			19	DD	00019		PUSHL	#25
00000000G	EF	00000000G	EF	9F	0001B		PUSHAB	FDL_DEST
			03	FB	00021		CALLS	#3,PASSCLOSE2
		000000FC	8F	DD	00028	1\$:	PUSHL	#252 ; 1199
			07	DD	0002E		PUSHL	#7
			04	DD	00030		PUSHL	#4
		00000000G	EF	9F	00032		PUSHAB	SY\$OUTPUT_NAME
			0B	DD	00038		PUSHL	#11
			01	DD	0003A		PUSHL	#1

Generated Code

1 2
16-Sep-1984 01:05:40
5-Sep-1984 13:38:00

00000000G	EF	00000000G	EF	9F 00042	PUSHAB	FDL DEST	:	1414
	01		01	FB 00048	CALLS	#1,PASSWRITELN2		
				04 0004F	RET			1416

; Routine Size: 80 bytes, Routine Base: \$CODE + 01BFB

				0004 00000	.ENTRY	SHOW CUR_PRI_SEC, ^M<R2>	:	1468
SE		04		C2 00002	SUBL2	#4, SP		
5C		BC		90 00005	MOVB	@4(R12), PLUS_VALUE		
	04			EF 04 00009	CLRL	LINES SHOWN	:	1475
50	00000000G	EF		D0 0000F	MOVL	DEF_CURRENT, R0	:	1482
52	00000000G	EF		90 00016	MOVB	(R0), SAVE_OBJECT_TYPE		
50	00000000G	EF		D0 00019	MOVL	DEF_CURRENT, R0	:	1484
				60 94 00020	CLRB	(R0)		
FC	AD			5C 90 00022	MOVB	PLUS_VALUE, -4(FP)	:	1485
		FC		AD 9F 00026	PUSHAB	-4(FP)		
14C5	CF			01 FB 00029	CALLS	#1, SHOW_CURRENT		
	50	00000000G		EF D0 0002E	MOVL	DEF_CURRENT, R0	:	1487
	OF	19		A0 91 00035	CMPB	25(R0), #15		
				00V 13 00039	BEQL	2\$		
	50	00000000G		EF D0 0003B	MOVL	DEF_CURRENT, R0	:	1491
	60			01 90 00042	MOVB	#1, (R0)		
FC	AD			5C 90 00045	MOVB	PLUS_VALUE, -4(FP)	:	1492
		FC		AD 9F 00049	PUSHAB	-4(FP)		
14C5	CF			01 FB 0004C	CALLS	#1, SHOW_CURRENT		
	50	00000000G		EF D0 00051	MOVL	DEF_CURRENT, R0	:	1496
	60			52 90 00058	MOVB	SAVE_OBJECT_TYPE, (R0)		
				04 0005B	RET			1498

; Routine Size: 92 bytes, Routine Base: \$CODE + 01C4B

01CA7 .END

COMMAND QUALIFIERS

PASCAL/MACHINE/NODEBUG/NOCHECK/LIS=LIS\$:EDFSHOW/OBJ=OBJ\$:EDFSHOW MSRC\$:EDFSHOW
/CHECK=(NOBOUNDS, NOCASE_SELECTOR, NOOVERFLOW, NOPOINTERS, NOSUBRANGE)
/DEBUG=(NOSYMBOL\$, NOTRACEBACK)
/ENVIRONMENT= \$255\$DUA28:[EDF.OBJ]EDFSHOW.PEN;1
/LIST= \$255\$DUA28:[EDF.LIS]EDFSHOW.LIS;1
/OBJECT= \$255\$DUA28:[EDF.OBJ]EDFSHOW.OBJ;1
/NOCROSS_REFERENCE /ERROR_LIMIT=30 /NOG_FLOATING /MACHINE_CODE /NOOLD_VERSION /OPTIMIZE /NOSTANDARD /WARNINGS

COMPILER INTERNAL TIMING

Phase	Faults	CPU Time	Elapsed Time
Initialization	83	00:00.4	00:02.5
Source Analysis	675	00:15.7	03:51.6
Source Listing	41	00:01.9	00:04.0
Tree Construction	185	00:01.0	00:02.1
Flow Analysis	39	00:00.3	00:00.5
Profit Analysis	30	00:00.5	00:01.1
Context Analysis	693	00:07.8	00:17.1
Name Packing	10	00:00.2	00:00.5
Code Selection	63	00:01.1	00:02.7
Final	361	00:06.2	00:16.8
TOTAL	2184	00:35.3	04:39.2

COMPILATION STATISTICS

CPU Time: 00:35.3 (2554 Lines/Minute)
Elapsed Time: 04:39.2
Page Faults: 2184
Compilation Complete

0127 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

The image displays a large grid of approximately 100 small terminal windows, each containing text-based data. The windows are arranged in a regular pattern across the page. Several windows are clearly legible and contain the following text:

- EDFSOLMSG LIS**: Located in the upper right quadrant.
- EDFMAIN LIS**: Located in the middle right quadrant.
- EDFSHOW LIS**: Located in the middle right quadrant, below EDFMAIN LIS.
- EDFMSG LIS**: Located in the lower right quadrant.
- EDFGRF LIS**: Located in the lower left quadrant.

The remaining windows contain various columns of text, some appearing to be lists or data tables, though they are too small to read in detail. The overall appearance is that of a multi-processor terminal session or a printout of a large data set.

The image displays a grid of 100 small terminal windows, each showing a different screen from the VAX/VMS operating system. The screens are arranged in a 10x10 grid. Many of the screens show system utilities and data listings. Some of the visible text includes:

- EDFTABLES LIS
- EDFUTIL LIS
- EDFTERM LIS
- EDFSTRUCT LIS
- ADMTITLE
- EDFUTIL
- EDFTERM
- EDFSTRUCT

The text is small and arranged in columns and rows, typical of a multi-processor terminal display.