

EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD	TTTTTTTTTTTTTTTT
EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD	TTTTTTTTTTTTTTTT
EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD	TTTTTTTTTTTTTTTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEEEEEEEEEEEEE	DDD	TTT
EEEEEEEEEEEEEE	DDD	TTT
EEEEEEEEEEEEEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD	TTT
EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD	TTT
EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD	TTT

```

LL      QQQQQQ    UU      UU      EEEEEEEEEEE RRRRRRRR    YY      YY
LL      QQQQQQ    UU      UU      EEEEEEEEEEE RRRRRRRR    YY      YY
LL      QQ        QQ    UU      UU      EE        RR          YY      YY
LL      QQ        QQ    UU      UU      EE        RR          YY      YY
LL      QQ        QQ    UU      UU      EE        RR          YY      YY
LL      QQ        QQ    UU      UU      EE        RR          YY      YY
LL      QQ        QQ    UU      UU      EE        RR          YY      YY
LL      QQ        QQ    UU      UU      EE        RR          YY      YY
LL      QQ        QQ    UU      UU      EE        RR          YY      YY
LL      QQ        QQ    UU      UU      EE        RR          YY      YY
LL      QQ        QQ    UU      UU      EE        RR          YY      YY
LL      QQ        QQ    UU      UU      EE        RR          YY      YY
LL      QQ        QQ    UU      UU      EE        RR          YY      YY
LLLLLLLLLLLLL  QQQQ    QQ    UUUUUUUUUU  EEEEEEEEEEE RR          RR    YY
LLLLLLLLLLLLL  QQQQ    QQ    UUUUUUUUUU  EEEEEEEEEEE RR          RR    YY

```

```

LL          IIIIII          SSSSSSSS
LL          IIIIII          SSSSSSSS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SSSSSS
LL          II             SSSSSS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SS
LLLLLLLLLLLL IIIIII          SSSSSSSS
LLLLLLLLLLLL IIIIII          SSSSSSSS

```



```
0001 0 %TITLE 'EDT$LQUERY - do /QUERY processing'
0002 0 MODULE EDT$LQUERY (
0003 0 IDENT = 'V04-000'
0004 0 ) =
0005 1 BEGIN
0006 1
0007 1 *****
0008 1 *
0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0011 1 * ALL RIGHTS RESERVED.
0012 1 *
0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0018 1 * TRANSFERRED.
0019 1 *
0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0022 1 * CORPORATION.
0023 1 *
0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0026 1 *
0027 1 *
0028 1 *****
0029 1
0030 1
0031 1 ++
0032 1 FACILITY: EDT -- The DEC Standard Editor
0033 1
0034 1 ABSTRACT:
0035 1
0036 1 This module handles the user interface of the /QUERY
0037 1 option on various line mode commands, such as SUBSTITUTE.
0038 1
0039 1 ENVIRONMENT: Runs at any access mode - AST reentrant
0040 1
0041 1 AUTHOR: Bob Kushlis, CREATION DATE: February 3, 1978
0042 1
0043 1 MODIFIED BY:
0044 1
0045 1 1-001 - Original. DJS 02-FEB-1981. This module was created by
0046 1 extracting the routine EDT$$PROC_QRYQAL from the module EXEC.BLI.
0047 1 1-002 - Regularize headers. JBS 20-Mar-1981
0048 1 1-003 - Use the ASSERT macro and fix up some comments mangled by converting QUERY
0049 1 to EDT$$PROC_QRYQAL. JBS 01-Jun-1981
0050 1 1-004 - Revise journaling. JBS 22-Jun-1981
0051 1 1-005 - Use new message codes. JBS 04-Aug-1981
0052 1 1-006 - Make empty responses illegal. JBS 16-Aug-1981
0053 1 1-007 - Prompt from a global. JBS 23-Oct-1981
0054 1 1-008 - Use heap storage instead of the stack to hold the constructed
0055 1 line. JBS 27-Jan-1982
0056 1 1-009 - Add a missing dot. JBS 28-Jan-1982
0057 1 1-010 - Add EDT$$G_JOU_VALID. JBS 09-Apr-1982
```

EDTSLQUERY
V04-000

EDTSLQUERY - do /QUERY processing

E 7
16-Sep-1984 00:55:45
14-Sep-1984 12:23:39

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[EDT.SRC]LQUERY.BLI;1 Page 2 (1)

```
.. 58      0058 1 | 1-011 - Refresh the screen after a query. JBS 05-Jul-1982
.. 59      0059 1 | 1-012 - If the journal file ends at a query response, treat the response
.. 60      0060 1 | as 'Q', to return to command level. Note that the 'Q' must
.. 61      0061 1 | be journaled. JBS 10-Jul-1982
.. 62      0062 1 | 1-013 - Added logic to maintain EDT$G_TIN_OBUFPOS durring /RECOVERY mode.
.. 63      0063 1 | REM 10-Oct-1983
.. 64      0064 1 | --
.. 65      0065 1 |
```


EDTSLQUERY
V04-000

EDTSLQUERY - do /QUERY processing
Declarations

F 7
16-Sep-1984 00:55:45
14-Sep-1984 12:23:39

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[EDT.SRC]LQUERY.BLI;1
Page 3
(2)

```

: 67      0066 1 %SBTTL 'Declarations'
: 68      0067 1
: 69      0068 1 : TABLE OF CONTENTS:
: 70      0069 1
: 71      0070 1
: 72      0071 1 REQUIRE 'EDTSRC:TRAROUNAM';
: 73      0510 1
: 74      0511 1 FORWARD ROUTINE
: 75      0512 1     EDT$$PROC_QRYQAL;
: 76      0513 1
: 77      0514 1
: 78      0515 1 : INCLUDE FILES:
: 79      0516 1
: 80      0517 1
: 81      0518 1 REQUIRE 'EDTSRC:EDTREQ';
: 82      0653 1
: 83      0654 1
: 84      0655 1 : MACROS:
: 85      0656 1
: 86      0657 1 :     NONE
: 87      0658 1
: 88      0659 1 : EQUATED SYMBOLS:
: 89      0660 1
: 90      0661 1
: 91      0662 1 LITERAL
: 92      0663 1     QUERY_LEN = 80,
: 93      0664 1     LINE_LEN = LIN_FIXED_SIZE + 255;
: 94      0665 1
: 95      0666 1
: 96      0667 1 : OWN STORAGE:
: 97      0668 1
: 98      0669 1 :     NONE
: 99      0670 1
: 100     0671 1 : EXTERNAL REFERENCES:
: 101     0672 1
: 102     0673 1 :     In the routine
```

! Process the EDT\$\$PROC_QRYQAL qualifier

```
104 0674 1 %SBTTL 'EDT$$PROC_QRYQAL - do /QUERY processing'
105 0675 1
106 0676 1 GLOBAL ROUTINE EDT$$PROC_QRYQAL (
107 0677 1     WPTR,
108 0678 1     WEND
109 0679 1 ) =
110 0680 1
111 0681 1 ++
112 0682 1 FUNCTIONAL DESCRIPTION:
113 0683 1
114 0684 1     EDT$$PROC_QRYQAL processing routine. This routine is called before operating on
115 0685 1     each line by those commands which can take a /QUERY qualifier.
116 0686 1
117 0687 1     The options bits are checked to see if the /QUERY qualifier was used.
118 0688 1     If not the routine returns success immediately. If it was specified,
119 0689 1     then the user is prompted for verification. The actions for each
120 0690 1     possible answer are:
121 0691 1
122 0692 1         Y - Return a 1 to indicate operation should be performed.
123 0693 1         N - Return a 0 to indicate operation should not be performed.
124 0694 1         Q - The global flag EDT$$G_EXE_QRYQUIT is set to stop further processing.
125 0695 1         A - The /QUERY option bit is cleared so no more queries are done.
126 0696 1
127 0697 1     If the answer is not one of the above, then a message is displayed and the
128 0698 1     user is prompted again.
129 0699 1
130 0700 1 FORMAL PARAMETERS:
131 0701 1
132 0702 1     WPTR                Used to reconstruct line for SUBSTITUTE, 0 otherwise
133 0703 1
134 0704 1     WEND                Used to reconstruct line for SUBSTITUTE, 0 otherwise
135 0705 1
136 0706 1 IMPLICIT INPUTS:
137 0707 1
138 0708 1     EDT$$G_RCOV_MOD
139 0709 1     EDT$$T_LN_BOF
140 0710 1     EDT$$G_LN_LEN
141 0711 1     EDT$$A_WK_LN
142 0712 1     EDT$$G_EXE_QRYQUIT
143 0713 1     EDT$$G_EXE_SBITS
144 0714 1     EDT$$T_PMT_QUERY
145 0715 1
146 0716 1 IMPLICIT OUTPUTS:
147 0717 1
148 0718 1     EDT$$G_JOU_VALID
149 0719 1     EDT$$G_TIN_OBUFPOS
150 0720 1
151 0721 1 ROUTINE VALUE:
152 0722 1
153 0723 1     1 = do the operation
154 0724 1     0 = don't do the operation
155 0725 1
156 0726 1 SIDE EFFECTS:
157 0727 1
158 0728 1     NONE
159 0729 1
160 0730 1 --
```



```
161 0731 1
162 0732 2 BEGIN
163 0733 3
164 0734 3 EXTERNAL ROUTINE
165 0735 3 EDTSSFMT_MSG, | Place the text of a message in the format buffer
166 0736 3 EDTSSRD_CMDLN, | Read a command line
167 0737 3 EDTSSRD_JOUTXT, | Read a text record from the journal file
168 0738 3 EDT$STI_FLUSHJOUFI : NOVALUE, | Write current journal buffer on journal file
169 0739 3 EDT$STI_BUFCH : NOVALUE, | Store a character in the journal buffer
170 0740 3 EDT$SCNV_UPC, | Convert to upper case
171 0741 3 EDT$STY_CURLN, | Display the current line
172 0742 3 EDT$SALO_HEAP, | Allocate heap storage
173 0743 3 EDT$SDEA_HEAP : NOVALUE, | Deallocate heap storage
174 0744 3 EDTSSFMT_CRLF; | Terminate a formatted line
175 0745 3
176 0746 3 EXTERNAL
177 0747 3 EDT$SG_RCOV_MOD,
178 0748 3 EDT$ST_LN_BUF : VECTOR [255, BYTE],
179 0749 3 EDT$SG_LN_LEN,
180 0750 3 EDT$SA_WK_LN : REF LIN_BLOCK,
181 0751 3 EDT$SG_EXE_QRYQUIT, | Quit flag for /QUERY operations.
182 0752 3 EDT$SG_EXE_SBITS, | The options switches.
183 0753 3 EDT$ST_PMT_QUERY : VECTOR [, BYTE], | Counted ASCII string for /QUERY prompt
184 0754 3 EDT$SG_JOU_VALID, | 1 = journal record is valid
185 0755 3 EDT$SG_TIN_OBUFPOS, | Position in journal output buffer
186 0756 3 EDT$SG_FMT_BOT; | We are printing at the bottom of the screen
187 0757 3
188 0758 3 MESSAGES ((PLSANSYNQ, INSMEM));
189 0759 3
190 0760 3 LOCAL
191 0761 3 QUERY_RESP : REF BLOCK [CH$ALLOCATION (QUERY_LEN)],
192 0762 3 QUERY_RESP_COMPLETE, | 1 = response is complete
193 0763 3 RET_VAL, | Return value
194 0764 3 LEN,
195 0765 3 SW_LINE, | Save EDT$SA_WK_LN
196 0766 3 IN, | Input char ptr
197 0767 3 OUT, | Output char ptr
198 0768 3 WLEN, | Length of rest of line
199 0769 3 L_WPTR, | Local copy of WPTR
200 0770 3 T_LINE : REF LIN_BLOCK; | Current line with substitutions
201 0771 3
202 0772 3 !+
203 0773 3 !- Check for the /QUERY bit. If it is clear then return 1.
204 0774 3 !-
205 0775 3
206 0776 3 IF .EDT$SG_EXE_SBITS<OPB_QUERY>
207 0777 3 THEN
208 0778 3 BEGIN
209 0779 3 !+
210 0780 3 !- Display the line so the user can see what he is verifying.
211 0781 3 !-
212 0782 3
213 0783 3 IF (.WPTR EQL 0)
214 0784 3 THEN
215 0785 3 EDT$STY_CURLN ()
216 0786 3 ELSE
217 0787 3 !+
```

```
218 0788 3 ! During a SUBSTITUTE command, the current line is in various
219 0789 3 ! pieces which are here reconstructed so that any substitution
220 0790 3 ! already made on the line are shown.
221 0791 3 !-
222 0792 4 BEGIN
223 0793 4 !+
224 0794 4 ! Allocate enough space for a maximum-length line.
225 0795 4 !-
226 0796 4
227 0797 4 IF EDT$$ALO_HEAP (%REF (LINE_LEN), T_LINE)
228 0798 4 THEN
229 0799 5 BEGIN
230 0800 5 !+
231 0801 5 ! Initialize the description for the line to be constructed.
232 0802 5 !-
233 0803 5 EDT$$CPY_MEM (LIN_FIXED_SIZE, .EDT$$A_WK_LN, .T_LINE);
234 0804 5 !+
235 0805 5 ! Copy the line up to the last substitution.
236 0806 5 !-
237 0807 5 IN = CH$PTR (EDT$$T_LN_BUF);
238 0808 5 OUT = CH$PTR (T_LINE [CIN_TEXT]);
239 0809 5
240 0810 5 DECR I FROM .EDT$$G_LN_LEN - 1 TO 0 DO
241 0811 5 CH$WCHAR_A (CH$RCHAR_A (IN), OUT);
242 0812 5
243 0813 5 !+
244 0814 5 ! Copy the current line from the last match to the end.
245 0815 5 !-
246 0816 5 WLEN = CH$DIFF (.WEND, .WPTR);
247 0817 5
248 0818 5 IF ((.EDT$$G_LN_LEN + .WLEN) GTR 255) THEN WLEN = MAX (0, 255 - .EDT$$G_LN_LEN);
249 0819 5
250 0820 5 L_WPTR = .WPTR;
251 0821 5
252 0822 5 DECR I FROM .WLEN - 1 TO 0 DO
253 0823 5 CH$WCHAR_A (CH$RCHAR_A (L_WPTR), OUT);
254 0824 5
255 0825 5 !+
256 0826 5 ! Fixup the description of the fake current line.
257 0827 5 !-
258 0828 5 T_LINE [LIN_LENGTH] = .EDT$$G_LN_LEN + .WLEN;
259 0829 5 !+
260 0830 5 ! Type the line.
261 0831 5 !-
262 0832 5 SW_LINE = .EDT$$A_WK_LN; ! Save the current line description
263 0833 5 EDT$$A_WK_LN = .T_LINE; ! Make the constructed line the current one
264 0834 5 EDT$$TY_CURLN (); ! Format and output this line
265 0835 5 EDT$$A_WK_LN = .SW_LINE; ! Restore the current line description
266 0836 5 !+
267 0837 5 ! Deallocate the heap storage used to hold the line.
268 0838 5 !-
269 0839 5 EDT$$DEA_HEAP (%REF (LINE_LEN), T_LINE);
270 0840 5 END
271 0841 4 ELSE
272 0842 5 BEGIN
273 0843 5 !+
274 0844 5 ! There is not enough heap storage to print the line. Don't do this operation
```



```
275 0845 5 ! and stop the whole command. Also, give an appropriate error message.
276 0846 5 !-
277 0847 5 EDT$$FMT_MSG (EDT$_INSMEM); ! Give an appropriate error message
278 0848 5 EDT$$G_EXE_QRYQUIT = 1; ! Stop the command
279 0849 5 RETURN(0); ! Don't do this substitution
280 0850 4 END;
281 0851 4
282 0852 3 END;
283 0853 3
284 0854 3 !+
285 0855 3 ! Allocate space to hold the response to the query.
286 0856 3 !-
287 0857 3
288 0858 4 IF ( NOT EDT$$ALO_HEAP (%REF (QUERY_LEN), QUERY_RESP))
289 0859 3 THEN
290 0860 4 BEGIN
291 0861 4 !+
292 0862 4 ! There is not enough storage to accept the response. Don't do this
293 0863 4 ! operation and stop the whole command. Also, give an appropriate error message.
294 0864 4 !-
295 0865 4 EDT$$FMT_MSG (EDT$_INSMEM); ! Give an appropriate message
296 0866 4 EDT$$G_EXE_QRYQUIT = 1; ! Stop the command
297 0867 4 RETURN(0); ! Don't do this substitution
298 0868 3 END;
299 0869 3
300 0870 3 QUERY_RESP_COMPLETE = 0;
301 0871 3
302 0872 3 WHILE ( NOT .QUERY_RESP_COMPLETE) DO
303 0873 4 BEGIN
304 0874 4 !+
305 0875 4 ! Get the line from either the terminal or the journal file.
306 0876 4 !-
307 0877 4
308 0878 4 IF .EDT$$G_RCOV_MOD
309 0879 4 THEN
310 0880 5 BEGIN
311 0881 5
312 0882 6 IF ( NOT EDT$$RD_JOUTXT (.QUERY_RESP, LEN))
313 0883 5 THEN
314 0884 6 BEGIN
315 0885 6 !+
316 0886 6 ! We have reached the end of the journal file. Fake a 'Q' response so that
317 0887 6 ! we will terminate this command without making any more changes to the buffer.
318 0888 6 !-
319 0889 6 LEN = 1;
320 0890 6 CH$WCHAR (%C'Q', .QUERY_RESP);
321 0891 6 !+
322 0892 6 ! We must journal the fake response, in case we do another /RECOVER during this session.
323 0893 6 !-
324 0894 6 EDT$$TI_BUFCH (CH$RCHAR (.QUERY_RESP));
325 0895 6 EDT$$G_JOU_VALID = 1
326 0896 6 END
327 0897 5 ELSE
328 0898 5 EDT$$G_TIN_OBUFPOS = MIN(.len, 1)
329 0899 5 END
330 0900 4 ELSE
331 0901 5 BEGIN
```

```
332 0902 5 +
333 0903 5 | Make sure the journal buffer has been written to the journal file,
334 0904 5 | since we are about to wait for terminal input.
335 0905 5 -
336 0906 5 | EDT$$TI_FLUSHJOUFI (%C'T');
337 0907 5 +
338 0908 5 | If all the text is being concentrated at the bottom of the screen, then be sure we are prompting
339 0909 5 | on a blank line.
340 0910 5 -
341 0911 5
342 0912 5 | IF .EDT$$G_FMT_BOT THEN EDT$$FMT_CRLF ();
343 0913 5
344 0914 5 | EDT$$RD_CMDLN (EDT$$T_PMT_QUERY [1], .EDT$$T_PMT_QUERY [0], .QUERY_RESP, LEN, QUERY_LEN);
345 0915 5 +
346 0916 5 | Make sure the response is journaled. Only the first character of the response is journaled
347 0917 5 | because only the first character is important.
348 0918 5 -
349 0919 5
350 0920 5 | IF (.LEN GEQ 1) THEN EDT$$TI_BUFCH (CH$RCHAR (.QUERY_RESP));
351 0921 5
352 0922 5 | EDT$$G_JOU_VALID = 1;
353 0923 4 | END;
354 0924 4
355 0925 4 +
356 0926 4 | Check out the answer.
357 0927 4 -
358 0928 4
359 0929 5 | IF (.LEN LSS 1)
360 0930 4 | THEN
361 0931 4 | EDT$$FMT_MSG (EDT$ _PLSANSYNQ)
362 0932 4 | ELSE
363 0933 5 | BEGIN
364 0934 5 | EDT$$CNV_UPC (.QUERY_RESP, 1);
365 0935 5 |
366 0936 5 | SELECT ONE CH$RCHAR (.QUERY_RESP) OF
367 0937 5 | SET
368 0938 5 |
369 0939 5 | [%C'Y'] :
370 0940 6 | BEGIN
371 0941 6 | RET_VAL = 1;
372 0942 6 | QUERY_RESP_COMPLETE = 1;
373 0943 5 | END;
374 0944 5 |
375 0945 5 | [%C'N'] :
376 0946 6 | BEGIN
377 0947 6 | RET_VAL = 0;
378 0948 6 | QUERY_RESP_COMPLETE = 1;
379 0949 5 | END;
380 0950 5 |
381 0951 5 | [%C'A'] :
382 0952 6 | BEGIN
383 0953 6 | EDT$$G_EXE_SBITS<OPB_QUERY> = 0;
384 0954 6 | RET_VAL = 1;
385 0955 6 | QUERY_RESP_COMPLETE = 1;
386 0956 5 | END;
387 0957 5 |
388 0958 5 | [%C'Q'] :
```



```
389 0959 6 BEGIN
390 0960 6 EDT$$G_EXE_QRYQUIT = 1;
391 0961 6 RET_VAL = 0;
392 0962 6 QUERY_RESP_COMPLETE = 1;
393 0963 6 END;
394 0964 6
395 0965 6 [OTHERWISE] :
396 0966 6 EDT$$FMT_MSG (EDT$_PLSANSYNQ);
397 0967 6 TES;
398 0968 6
399 0969 6 END;
400 0970 6
401 0971 6 END;
402 0972 6
403 0973 6
404 0974 6 + Come here when the query response is complete. RET_VAL contains the
405 0975 6 value to return. Deallocate the heap storage used to hold the responses
406 0976 6 to the query.
407 0977 6
408 0978 6 EDT$$DEA_HEAP (%REF (QUERY_LEN), QUERY_RESP);
409 0979 6 END
410 0980 6 ELSE
411 0981 6 RET_VAL = 1;
412 0982 6
413 0983 6 RETURN (.RET_VAL);
414 0984 6 END;
```

! of routine EDT\$\$PROC_QRYQAL

```
.TITLE EDT$LQUERY EDT$LQUERY - do /QUERY processing
.IDENT \V04-000\
```

```
.EXTRN EDT$$FMT_MSG, EDT$$RD_CMDLN
.EXTRN EDT$$RD_JOUTXT, EDT$$TI_FLUSHJOUFI
.EXTRN EDT$$TI_BUFCH, EDT$$CNV_UPC
.EXTRN EDT$$TY_CURLN, EDT$$ALO_HEAP
.EXTRN EDT$$DEA_HEAP, EDT$$FMT_CRLF
.EXTRN EDT$$G_RCOV_MOD
.EXTRN EDT$$T_LN_BUF, EDT$$G_LN_LEN
.EXTRN EDT$$A_WK_LN, EDT$$G_EXE_QRYQUIT
.EXTRN EDT$$G_EXE_SBITS
.EXTRN EDT$$T_PMT_QUERY
.EXTRN EDT$$G_JOU_VALID
.EXTRN EDT$$G_TIN_OBUFPOS
.EXTRN EDT$$G_FMT_BOT, EDT$_PLSANSYNQ
.EXTRN EDT$_INSMEM
```

```
.PSECT _EDT$CODE, NOWRT, SHR, PIC, 2
```

```
.ENTRY EDT$$PROC_QRYQAL, Save R2,R3,R4,R5,R6,R7,- ; 0676
R8,R9,R10,R11
MOVAB EDT$$DEA_HEAP, R11
MOVAB EDT$$ALO_HEAP, R10
MOVAB EDT$$TY_CURLN, R9
MOVAB EDT$$A_WK_LN, R8
SUBL2 #16, SP
BBS #1, EDT$$G_EXE_SBITS, 1$ ; 0776
BRW 30$
```

OFFC 00000

```
5B 00000000G 00 9E 00002
5A 00000000G 00 9E 00009
59 00000000G 00 9E 00010
58 00000000G 00 9E 00017
5E 10 C2 0001E
03 00000000G 00 01 E0 00021
01B0 31 00029
```

	56	04	AC	D0	0002C	1\$:	MOVL	WPTR, R6		0783
			06	12	00030		BNEQ	2\$		
	69		00	FB	00032		CALLS	#0, EDT\$\$TY_CURLN		0785
			008B	31	00035		BRW	10\$		
		04	AE	9F	00038	2\$:	PUSHAB	T LINE		0797
	04	AE	0106	8F	3C		MOVZWL	#262, 4(SP)		
		04	AE	9F	00041		PUSHAB	4(SP)		
	6A		02	FB	00044		CALLS	#2, EDT\$\$ALO_HEAP		
	03		50	E8	00047		BLBS	R0, 3\$		
			0087	31	0004A		BRW	11\$		
	57		68	D0	0004D	3\$:	MOVL	EDT\$\$A_WK_LN, R7		0803
04	BE		67	07	28		MOVCL	#7, (R7), @T_LINE		
			51	00	9E		MOVAB	EDT\$\$T_LN_BUF, IN		0807
	52	04	AE	07	C1		ADDL3	#7, T LINE, OUT		0808
			53	00	D0		MOVL	EDT\$\$G_LN_LEN, R3		0810
			50	53	D0		MOVL	R3, I		0811
				03	11		BRB	5\$		
	82		81	90	0006D	4\$:	MOVB	(IN)+, (OUT)+		
	FA		50	F4	00070	5\$:	SOBGEQ	I, 4\$		
		08	AC	56	C3		SUBL3	R6, WEND, WLEN		0816
	50		53	50	C1		ADDL3	WLEN, R3, R1		0818
	51		8F	51	D1		CMPL	R1, #255		
				0F	15		BLEQ	7\$		
	51		8F	53	C3		SUBL3	R3, #255, R1		
				02	18		BGEQ	6\$		
				51	D4		CLRL	R1		
	50		51	D0	00091	6\$:	MOVL	R1, WLEN		
	54		56	D0	00094	7\$:	MOVL	R6, L_WPTR		0820
	51		50	D0	00097		MOVL	WLEN, I		0823
				03	11		BRB	9\$		
	82		84	90	0009C	8\$:	MOVB	(L_WPTR)+, (OUT)+		
	FA		51	F4	0009F	9\$:	SOBGEQ	I, 8\$		
04	BE		53	50	81		ADDB3	WLEN, R3, @T_LINE		0828
			52	57	D0		MOVL	R7, SW_LINE		0832
			68	04	AE		NOVL	T LINE, EDT\$\$A_WK_LN		0833
			69	00	FB		CALLS	#0, EDT\$\$TY_CURLN		0834
			68	52	D0		MOVL	SW LINE, EDT\$\$A_WK_LN		0835
				AE	9F		PUSHAB	T LINE		0839
		04	AE	0106	8F		MOVZWL	#262, 4(SP)		
			04	AE	9F		PUSHAB	4(SP)		
	6B		02	FB	000C0		CALLS	#2, EDT\$\$DEA_HEAP		
			AE	9F	000C3	10\$:	PUSHAB	QUERY_RESP		0858
	04	AE	50	8F	9A		MOVZBL	#80, 4(SP)		
			04	AE	9F		PUSHAB	4(SP)		
	6A		02	FB	000CE		CALLS	#2, EDT\$\$ALO_HEAP		
	17		50	E8	000D1		BLBS	R0, 12\$		
			8F	DD	000D4	11\$:	PUSHL	#EDT\$ INSMEM		0865
	00000000G		00	01	FB		CALLS	#1, EDT\$\$FMT MSG		
	00000000G		00	01	D0		MOVL	#1, EDT\$\$G_EXE_QRYQUIT		0866
				00F8	31		BRW	32\$		0867
				53	D4		CLRL	QUERY_RESP_COMPLETE		0870
	52		0C	AE	D0		MOVL	QUERY_RESP, R2		0882
	03		53	E9	000F1	13\$:	BLBC	QUERY_RESP_COMPLETE, 14\$		0872
				00D5	31		BRW	29\$		
	2E		00000000G	00	E9		BLBC	EDT\$\$G_RCOV_MOD, 17\$		0878
			08	AE	9F		PUSHAB	LEN		0882
				52	DD		PUSHL	R2		

00000000G	00	02	FB	00103	CALLS	#2, EDT\$SRD_JOUTXT	
	0A	50	E8	0010A	BLBS	R0, 15\$	
08	AE	01	D0	0010D	MOVL	#1, LEN	0889
	62	51	8F	90 00111	MOVB	#81, (R2)	0890
			50	11 00115	BRB	19\$	0894
	50	08	AE	D0 00117	MOVL	LEN, R0	0898
	01		50	D1 0011B	CMPL	R0, #1	
			03	15 0011E	BLEQ	16\$	
	50		01	D0 00120	MOVL	#1, R0	
00000000G	00		50	D0 00123	MOVL	R0, EDT\$G_TIN_OBUFPOS	0880
			4C	11 0012A	BRB	21\$	0906
	7E	54	8F	9A 0012C	MOVZBL	#84, -(SP)	
00000000G	00		01	FB 00130	CALLS	#1, EDT\$STI_FLUSHJOUFI	
	07	00000000G	00	E9 00137	BLBC	EDT\$G_FMT BOT, 18\$	0912
00000000G	00		00	FB 0013E	CALLS	#0, EDT\$FMT_CRLF	
	7E	50	8F	9A 00145	MOVZBL	#80, -(SP)	0914
		0C	AE	9F 00149	PUSHAB	LEN	
			52	DD 0014C	PUSHL	R2	
	7E	00000000G	00	9A 0014E	MOVZBL	EDT\$ST_PMT_QUERY, -(SP)	
		00000000G	00	9F 00155	PUSHAB	EDT\$ST_PMT_QUERY+1	
00000000G	00		05	FB 0015B	CALLS	#5, EDT\$SRD_CMDLN	
		08	AE	D5 00162	TSTL	LEN	0920
			0A	15 00165	BLEQ	20\$	
	7E		62	9A 00167	MOVZBL	(R2), -(SP)	
00000000G	00		01	FB 0016A	CALLS	#1, EDT\$STI_BUFCH	
00000000G	00		01	D0 00171	MOVL	#1, EDT\$G_JOU_VALID	0922
		08	AE	D5 00178	TSTL	LEN	0929
			3F	15 0017B	BLEQ	27\$	
			01	DD 0017D	PUSHL	#1	0934
			52	DD 0017F	PUSHL	R2	
00000000G	00		02	FB 00181	CALLS	#2, EDT\$CNV_UPC	
59	8F		62	91 00188	CMPB	(R2), #89	0939
			05	12 0018C	BNEQ	23\$	
	54		01	D0 0018E	MOVL	#1, RET_VAL	0941
			24	11 00191	BRB	26\$	0942
4E	8F		62	91 00193	CMPB	(R2), #78	0945
			1C	13 00197	BEQL	25\$	
41	8F		62	91 00199	CMPB	(R2), #65	0951
			09	12 0019D	BNEQ	24\$	
00000000G	00		02	8A 0019F	BICB2	#2, EDT\$G_EXE_SBITS	0953
			E6	11 001A6	BRB	22\$	0954
51	8F		62	91 001A8	CMPB	(R2), #81	0958
			0E	12 001AC	BNEQ	27\$	
00000000G	00		01	D0 001AE	MOVL	#1, EDT\$G_EXE_QRYQUIT	0960
			54	D4 001B5	CLRL	RET_VAL	0961
	53		01	D0 001B7	MOVL	#1, QUERY_RESP_COMPLETE	0962
			0D	11 001BA	BRB	28\$	0936
		00000000G	8F	DD 001BC	PUSHL	#EDT\$ PLSANSYNQ	0966
00000000G	00		01	FB 001C2	CALLS	#1, EDT\$FMT_MSG	
		FF	25	31 001C9	BRW	13\$	0872
		0C	AE	9F 001CC	PUSHAB	QUERY_RESP	0978
		50	8F	9A 001CF	MOVZBL	#80, 4(SP)	
04	AE	04	AE	9F 001D4	PUSHAB	4(SP)	
			02	FB 001D7	CALLS	#2, EDT\$DEA_HEAP	
			03	11 001DA	BRB	31\$	0776
	54		01	D0 001DC	MOVL	#1, RET_VAL	0981
	50		54	D0 001DF	MOVL	RET_VAL, R0	0983

```

: 415      0985 1
: 416      0986 1 !<BLF/PAGE>

```


EDTSLQUERY
V04-000

EDTSLQUERY - do /QUERY processing
EDT\$\$PROC_QRYQAL - do /QUERY processing

C 8
16-Sep-1984 00:55:45
14-Sep-1984 12:23:39

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[EDT.SRC]LQUERY.BLI;1 Page 13
(4)

: 418 0987 1 END
: 419 0988 1
: 420 0989 0 ELUDOM

! of module EDTSLQUERY

PSECT SUMMARY

Name	Bytes	Attributes
_EDT\$CODE	486	NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[EDT.SRC]EDT.L32;1	377	10	2	40	00:00.2
_\$255\$DUA28:[EDT.SRC]PSECTS.L32;1	2	1	50	7	00:00.1

COMMAND QUALIFIERS

; BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACEBACK/LIS=LIS\$:LQUERY/OBJ=OBJ\$:LQUERY MSRC\$:LQUERY.BLI/UPDATE=(ENH\$:LQUERY)

; Size: 486 code + 0 data bytes
; Run Time: 00:22.7
; Elapsed Time: 00:27.2
; Lines/CPU Min: 2609
; Lexemes/CPU-Min: 8160
; Memory Used: 152 pages
; Compilation Complete

0136

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY