

🍏 Apple Lisa Computer  
Technical Information



# Apple Lisa Computer: Lisa File Label Reader Utility (David T Craig 1987)

Lisa Computer:  
1983 - 1985

File - UTIL/FILELABELS.RUN.TEXT < 16-DEC-1987 20:19:10 > Page - 0001

Apple Lisa FILE LABEL Reader Utility 1.00  
Not (c) December 6, 1987 by David T. Craig

#357

This program displays the contents of the LABEL area for a file. The label allows an application to keep file data separate from information maintained about the file. Labels can be used for any object in the Lisa File System and each label has a maximum size of 128 bytes.

Enter volume name (Press <Return> to Quit) ? -UPPER  
Enter file name prefix ? (T5

\*\*\*\*\*  
\*\*\*\* [Size: 128] LABEL FOR FILE "-UPPER-(T5)dbcPHRASE"  
\*\*\*\*\*

Addr	0	2	4	6	8	A	C	E	
[ 00 ]	:	0000	0000	0000	0000	0000	0000	0000	<.....>
[ 10 ]	:	0000	0000	0000	0000	0000	0000	0000	<.....>
[ 20 ]	:	0000	0000	0000	0000	0000	0000	0000	<.....>
[ 30 ]	:	0000	0000	0000	0000	0000	0000	0000	<.....>
[ 40 ]	:	0000	0000	0000	0000	0000	0000	0000	<.....>
[ 50 ]	:	0000	0000	0000	0000	0000	0000	0000	<.....>
[ 60 ]	:	0000	0000	0000	0000	0000	0000	0000	<.....>
[ 70 ]	:	0000	0000	0000	0000	0000	0000	0000	<.....>

\*\*\*\*\*  
\*\*\*\* [Size: 128] LABEL FOR FILE "-UPPER-(T5)lmpHRASE"  
\*\*\*\*\*

Addr	0	2	4	6	8	A	C	E	
[ 00 ]	:	0000	0000	0000	0000	0000	0000	0000	<.....>
[ 10 ]	:	0000	0000	0000	0000	0000	0000	0000	<.....>
[ 20 ]	:	0000	0000	0000	0000	0000	0000	0000	<.....>
[ 30 ]	:	0000	0000	0000	0000	0000	0000	0000	<.....>
[ 40 ]	:	0000	0000	0000	0000	0000	0000	0000	<.....>
[ 50 ]	:	0000	0000	0000	0000	0000	0000	0000	<.....>
[ 60 ]	:	0000	0000	0000	0000	0000	0000	0000	<.....>
[ 70 ]	:	0000	0000	0000	0000	0000	0000	0000	<.....>

\*\*\*\*\*  
\*\*\*\* [Size: 128] LABEL FOR FILE "-UPPER-(T5)OBJ"  
\*\*\*\*\*

Addr	0	2	4	6	8	A	C	E	
[ 00 ]	:	0002	084C	6973	614C	6973	7400	0000	<...LisaList....>
[ 10 ]	:	0000	0000	0000	0000	0000	0000	0000	<.....>
[ 20 ]	:	0000	0000	0000	0000	0000	0000	0000	<.....>
[ 30 ]	:	0000	0000	0000	0000	0000	0000	0000	<.....>
[ 40 ]	:	0000	0018	0000	0028	000A	0122	0280	<.....(....">
[ 50 ]	:	0000	0000	0000	0019	0000	0000	0000	<.....>
[ 60 ]	:	0000	0000	0000	0000	0000	0000	0000	<.....>
[ 70 ]	:	0000	0000	0000	0000	0000	0000	0000	<.....>

Enter volume name (Press <Return> to Quit) ?

That's all, Folks ...

See Toolkit sources  
for definition of  
label contents (search  
for TLabel)

(10 pages)

Lisa WorkShop Utility : FileLabels [David Craig]

File - UTIL/FILELABELS.SOURCE.TEXT

< 16-DEC-1987 20:23:45 > Page - 0001

[illegible]

**Lisa WorkShop Utility : FileLabels [David Craig]**

```

78 1 78 -- | OUTPUT : (none)
79 1 79 -- |=====|
80 1 80 --
81 1 81 -- A  PROCEDURE Omega;
82 1 82 --
83 1 83 0- A  BEGIN ( ----- Omega ----- )
84 1 84 --      WRITELN;
85 1 85 --      WRITELN('That's all, Folks ...');
86 1 86 0- A  END; ( ----- Omega ----- )
87 1 87 --
88 1 88 -- ( |=====|
89 1 89 -- | ROUTINE : FetchVolumeName
90 1 90 -- | PURPOSE : Prompt & fetch a volume name
91 1 91 -- | INPUT   : (none)
92 1 92 -- | OUTPUT  : volume_name - Name of volume
93 1 93 -- |          user_is_done - User termination flag (TRUE --> quit)
94 1 94 -- |=====| )
95 1 95 --
96 1 96 -- A  PROCEDURE FetchVolumeName(VAR volume_name : gt_VolumeName;
97 1 97 --      VAR user_is_done : BOOLEAN);
98 1 98 --
99 1 99 0- A  BEGIN ( ----- FetchVolumeName ----- )
100 1 100 --      WRITELN; ( Fetch the volume name )
101 1 101 --      WRITE('Enter volume name (Press <Return> to Quit) ? ');
102 1 102 --      READLN(volume_name);
103 1 103 --
104 1 104 --      user_is_done := (volume_name = ''); ( Test if user is done )
105 1 105 0- A  END; ( ----- FetchVolumeName ----- )
106 1 106 --
107 1 107 -- ( |=====|
108 1 108 -- | ROUTINE : ExtractFileLabels
109 1 109 -- | PURPOSE : Extract the file labels from all the volume files
110 1 110 -- | INPUT   : volume_name - Name of volume to extract from
111 1 111 -- | OUTPUT  : (none)
112 1 112 -- |=====| )
113 1 113 --
114 1 114 -- A  PROCEDURE ExtractFileLabels(volume_name : gt_VolumeName);
115 1 115 --
116 1 116 --      CONST
117 1 117 --          k_NoError      = 0; ( No error code value )
118 1 118 --          k_EndOfCatalog = 848; ( End of catalog error code )
119 1 119 --
120 1 120 --      TYPE
121 1 121 --          t_Phrase      = STRING[79]; ( A generic phrase )
122 1 122 --          t_Byte        = 0..255; ( Good old 8-bit byte )
123 1 123 --          t_LabelBuffer = PACKED ARRAY [0..127] OF t_Byte; ( Label buffer )
124 1 124 --          t_HexBString = STRING[2]; ( Hexadecimal Byte string )
125 1 125 --
126 1 126 --      VAR
127 1 127 --          error      : INTEGER; ( Error result )
128 1 128 --          entry_prefix : E_NAME; ( Catalog file entry prefix string )
129 1 129 --          entry_info  : Q_INFO; ( Catalog file entry information )
130 1 130 --          label_file  : PATHNAME; ( Catalog file label name )
131 1 131 --          label_buffer : t_LabelBuffer; ( Label buffer (128 bytes) )
132 1 132 --          label_size  : LONGINT; ( Actual size of label (<= 128 bytes) )
133 1 133 --          user_stop   : BOOLEAN; ( User stop flag (TRUE --> stop) )
134 1 134 --
135 1 135 --      ( -----
136 1 136 --      | Routine : PABORTFLAG
137 1 137 --      | Purpose : Test for Apple-. keypress
138 1 138 --      | Input   : (none)
139 1 139 --      | Output  : PABORTFLAG - TRUE --> Apple-. pressed
140 1 140 --      ----- )
141 1 141 --
142 1 142 -- B  FUNCTION PABORTFLAG : BOOLEAN; EXTERNAL; ( Built into Lisa PasLib )
143 1 143 --
144 1 144 --      ( -----
145 1 145 --      | Routine : ShowError
146 1 146 --      | Purpose : Display an error code and message
147 1 147 --      | Input   : err_code - Error code
148 1 148 --      |          err_msg  - Error message
149 1 149 --      | Output  : (none)
150 1 150 --      ----- )
151 1 151 --
152 1 152 -- B  PROCEDURE ShowError(err_code : INTEGER; err_msg : t_Phrase);
153 1 153 --
154 1 154 0- B  BEGIN ( ----- ShowError ----- )

```

```

155 1 155 --      WRITELN('*** ERROR # ',err_code:1, ' : ',err_msg);
156 1 156 --      END; ( ----- ShowError ----- )
157 1 157 --
158 1 158 --      ( -----
159 1 159 --      | Routine : byte_to_hex
160 1 160 --      | Purpose : Convert a byte into a 2 digit hexadecimal string
161 1 161 --      | Input   : the_byte - Byte value
162 1 162 --      | Output  : hex_str  - Hexadecimal string
163 1 163 --      | ----- )
164 1 164 --
165 1 165 --      B      PROCEDURE byte_to_hex(the_byte : t_Byte; VAR hex_str : t_HexBString);
166 1 166 --
167 1 167 --      TYPE
168 1 168 --      t_Nibble = 0..15; ( A nibble or half byte )
169 1 169 --
170 1 170 --      VAR
171 1 171 --      msn : t_Nibble; ( Most significant nibble of byte )
172 1 172 --      lsn : t_Nibble; ( Least significant nibble of byte )
173 1 173 --
174 1 174 --      ( ----- )
175 1 175 --
176 1 176 --      C      FUNCTION nibble_to_hex(the_nibble : t_Nibble) : CHAR;
177 1 177 --
178 1 178 --      VAR
179 1 179 --      hex_nibble : CHAR;
180 1 180 --
181 1 181 --      BEGIN ( ----- nibble_to_hex ----- )
182 1 182 --      CASE the_nibble OF
183 1 183 --      0 : hex_nibble := '0';      1 : hex_nibble := '1';
184 1 184 --      2 : hex_nibble := '2';      3 : hex_nibble := '3';
185 1 185 --      4 : hex_nibble := '4';      5 : hex_nibble := '5';
186 1 186 --      6 : hex_nibble := '6';      7 : hex_nibble := '7';
187 1 187 --      8 : hex_nibble := '8';      9 : hex_nibble := '9';
188 1 188 --      10 : hex_nibble := 'A';     11 : hex_nibble := 'A';
189 1 189 --      12 : hex_nibble := 'C';     13 : hex_nibble := 'D';
190 1 190 --      14 : hex_nibble := 'E';     15 : hex_nibble := 'F';
191 1 191 --      END;
192 1 192 --
193 1 193 --      nibble_to_hex := hex_nibble;
194 1 194 --      END; ( ----- nibble_to_hex ----- )
195 1 195 --
196 1 196 --      ( -----
197 1 197 --
198 1 198 --      B      BEGIN ( ----- byte_to_hex ----- )
199 1 199 --      msn := the_byte DIV 16;
200 1 200 --      lsn := the_byte MOD 16;
201 1 201 --
202 1 202 --      hex_str := 'ML';
203 1 203 --
204 1 204 --      hex_str[1] := nibble_to_hex(msn);
205 1 205 --      hex_str[2] := nibble_to_hex(lsn);
206 1 206 --      END; ( ----- byte_to_hex ----- )
207 1 207 --
208 1 208 --      ( -----
209 1 209 --      | Routine : DumpLabelData
210 1 210 --      | Purpose : Display the label contents as a hexadecimal table
211 1 211 --      | Input   : lbl_file_name - Label file name
212 1 212 --      |           lbl_size      - Size of label ( <= 128 bytes)
213 1 213 --      |           lbl_buffer    - Label buffer (128 bytes)
214 1 214 --      | Output  : (none)
215 1 215 --      | ----- )
216 1 216 --
217 1 217 --      B      PROCEDURE DumpLabelData(lbl_file_name : PATHNAME;
218 1 218 --      lbl_size : LONGINT;
219 1 219 --      lbl_buffer : t_LabelBuffer);
220 1 220 --
221 1 221 --      CONST
222 1 222 --      k_Divider = '*****'; ( 35 '*'s )
223 1 223 --
224 1 224 --      TYPE
225 1 225 --      t_AsciiData = STRING[16];
226 1 226 --
227 1 227 --      VAR
228 1 228 --      ascii_data : t_AsciiData;
229 1 229 --      row : 0..7;
230 1 230 --      column : 0..15;
231 1 231 --      hex_str : t_HexBString;

```

```

232 1 232 --      a_byte      : t_UByte;
233 1 233 --
234 1 234 0- B      BEGIN ( ----- DumpLabelData ----- )
235 1 235 --      ( Show label table title )
236 1 236 --
237 1 237 --      WRITELN;
238 1 238 --      WRITELN(k_Divider,k_Divider);
239 1 239 --      WRITELN('**** [Size: ',lbl_size:3,'] LABEL FOR FILE "',
240 1 240 --                               lbl_file_name,'"');
241 1 241 --      WRITELN(k_Divider,k_Divider);
242 1 242 --
243 1 243 --      ( Initialize the table Ascii area )
244 1 244 --
245 1 245 --      ascii_data := '0123456789abcdef';
246 1 246 --
247 1 247 --      ( Display the table address & byte offset header )
248 1 248 --
249 1 249 --      WRITELN;
250 1 250 --      WRITELN(' Addr      0      2      4      6      8      A      C      E');
251 1 251 --      WRITELN(' ----- ; ----- ');
252 1 252 --
253 1 253 --      ( Display the table contents row by row )
254 1 254 --
255 1 255 --      FOR row := 0 TO 7 DO
256 1 256 1-          BEGIN
257 1 257 --          byte_to_hex(row * 16 , hex_str);
258 1 258 --
259 1 259 --          WRITE(' [ ',hex_str,' ] :');
260 1 260 --
261 1 261 --          ( Display a table row )
262 1 262 --
263 1 263 --          FOR column := 0 TO 15 DO
264 1 264 2-              BEGIN
265 1 265 --              IF (column MOD 2 = 0) THEN
266 1 266 --                  WRITE(' ');
267 1 267 --
268 1 268 --                  a_byte := lbl_buffer[(row * 16) + column];
269 1 269 --
270 1 270 --                  byte_to_hex(a_byte , hex_str);
271 1 271 --
272 1 272 --                  WRITE(hex_str);
273 1 273 --
274 1 274 --                  ( Setup the table Ascii area )
275 1 275 --
276 1 276 --                  IF (a_byte >= 32) AND (a_byte <= 127) THEN
277 1 277 --                      ascii_data[column+1] := CHR(a_byte)
278 1 278 --                  ELSE
279 1 279 --                      ascii_data[column+1] := '.';
280 1 280 -2              END; { FOR column }
281 1 281 --
282 1 282 --          WRITELN(' <',ascii_data,'>');
283 1 283 -1          END; { FOR row }
284 1 284 0- B      END; ( ----- DumpLabelData ----- )
285 1 285 --
286 1 286 --      ( ----- )
287 1 287 --
288 1 288 0- A      BEGIN ( ----- ExtractFileLabels ----- )
289 1 289 --      ( Open specified catalog )
290 1 290 --
291 1 291 --      RESET_CATALOG(error,volume_name);
292 1 292 --
293 1 293 --      ( Handle any I/O errors )
294 1 294 --
295 1 295 --      IF (error <> k_NoError) THEN
296 1 296 --          ShowError(error,CONCAT('Reseting Catalog "',volume_name,'" failed'))
297 1 297 --      ELSE
298 1 298 1-          BEGIN
299 1 299 --          ( Prompt user for file name prefix (if needed) )
300 1 300 --
301 1 301 --          WRITE('Enter file name prefix ? ');
302 1 302 --          READLN(entry_prefix);
303 1 303 --
304 1 304 --          ( Access all the files with the specified name prefix in the )
305 1 305 --          ( specified volume and display these files' label area )
306 1 306 --
307 1 307 --          user_stop := FALSE; ( User can't stop before starting )
308 1 308 --

```

File - UTIL/FILELABELS.SOURCE.TEXT

< 16-DEC-1987 20:23:45 > Page - 0005

[illegible]

**Lisa WorkShop Utility : FileLabels [David Craig]**

File - UTIL/FILELABELS.SOURCE.TEXT

< 16-DEC-1987 20:23:45 > Page - 0006

[illegible]

**Lisa WorkShop Utility : FileLabels [David Craig]**



File - UTIL/FILELABELS.SOURCE.TEXT

< 16-DEC-1987 20:23:45 > Page - 0007

1. -LOWER-UTIL/FileLabels.TEXT

```

-A-
  alpha      60*( 1) 372 ( 1)
  ascii_data 228*( 1) 245=( 1) 277=( 1) 279=( 1) 282 ( 1)
  a_byte     232*( 1) 268=( 1) 270 ( 1) 276 ( 1) 276 ( 1) 277 ( 1)

-B-
  BOOLEAN    46 ( 1) 97 ( 1) 133 ( 1) 142 ( 1)
  byte_to_hex 165*( 1) 257 ( 1) 270 ( 1)

-C-
  CHAR       176 ( 1) 179 ( 1)
  CHR        277 ( 1)
  column     238*( 1) 263=( 1) 265 ( 1) 268 ( 1) 277 ( 1) 279 ( 1)
  CONCAT     296 ( 1) 329 ( 1) 342 ( 1)

-D-
  dumplabeldata 217*( 1) 349 ( 1)

-E-
  entry_info  129*( 1) 313 ( 1) 325 ( 1) 331 ( 1)
  entry_prefix 128*( 1) 302 ( 1) 313 ( 1)
  error       127*( 1) 291 ( 1) 295 ( 1) 296 ( 1) 313 ( 1) 317 ( 1) 319 ( 1)
              320 ( 1) 335 ( 1) 341 ( 1) 342 ( 1) 361 ( 1)
  err_code    152*( 1) 155 ( 1)
  err_msg     152*( 1) 155 ( 1)
  etype       325 ( 1)
  extractfilelabel 114*( 1) 385 ( 1)
  e_name      128 ( 1)

-F-
  FALSE       307 ( 1)
  fetchvolumename 96*( 1) 380 ( 1)
  fileentry   325 ( 1)

-G-
  gc_pgmauthor 38*( 1) 64 ( 1)
  gc_pgmdate   40*( 1) 64 ( 1)
  gc_pgversion 39*( 1) 63 ( 1)
  gt_volumename 43*( 1) 47 ( 1) 96 ( 1) 114 ( 1)
  gv_userisdone 46*( 1) 380 ( 1) 384 ( 1) 387 ( 1)
  gv_volumename 47*( 1) 380 ( 1) 385 ( 1)

-H-
  hex_nibble  179*( 1) 183=( 1) 183=( 1) 184=( 1) 184=( 1) 185=( 1) 185=( 1)
              186=( 1) 186=( 1) 187=( 1) 187=( 1) 188=( 1) 188=( 1) 189=( 1)
              189=( 1) 190=( 1) 190=( 1) 193 ( 1)
  hex_str     165*( 1) 202=( 1) 204=( 1) 205=( 1) 231*( 1) 257 ( 1) 259 ( 1)
              270 ( 1) 272 ( 1)

-I-
  INTEGER     127 ( 1) 152 ( 1)

-K-
  k_divider   222*( 1) 238 ( 1) 238 ( 1) 241 ( 1) 241 ( 1)
  k_endofcatalog 118*( 1) 317 ( 1)
  k_noerror   117*( 1) 295 ( 1) 319 ( 1) 341 ( 1) 361 ( 1)

-L-
  label_buffer 131*( 1) 337 ( 1) 351 ( 1)
  label_file   130*( 1) 329=( 1) 336 ( 1) 343 ( 1) 349 ( 1)
  label_size   132*( 1) 339 ( 1) 350 ( 1)
  lbl_buffer   219*( 1) 268 ( 1)
  lbl_file_name 217*( 1) 240 ( 1)
  lbl_size     218*( 1) 239 ( 1)
  lisa_file_labels 25*( 1)
  LONGINT      132 ( 1) 218 ( 1)
  lookup_next_entry 313 ( 1)
  lsn          172*( 1) 200=( 1) 205 ( 1)

-M-
  msn         171*( 1) 199=( 1) 204 ( 1)

-N-
  name        331 ( 1)
  nibble_to_hex 176*( 1) 193=( 1) 204 ( 1) 205 ( 1)

```

Lisa WorkShop Utility : FileLabels [David Craig]

File - UTIL/FILELABELS.SOURCE.TEXT

< 16-DEC-1987 20:23:45 > Page - 0008

```

-O-
  omega      81*( 1) 389 ( 1)
  ORD4      337 ( 1)

-P-
  pabortflag 142*( 1) 355 ( 1)
  pathname   43 ( 1) 130 ( 1) 217 ( 1)

-Q-
  q_info     129 ( 1)

-R-
  READLN     102 ( 1) 302 ( 1)
  read_label 335 ( 1)
  reset_catalog 291 ( 1)
  row        229*( 1) 255=( 1) 257 ( 1) 268 ( 1)

-S-
  showerror  152*( 1) 296 ( 1) 320 ( 1) 342 ( 1)
  STRING     121 ( 1) 124 ( 1) 225 ( 1)
  syscall    31*( 1)

-T-
  the_byte   165*( 1) 199 ( 1) 200 ( 1)
  the_nibble 176*( 1) 182 ( 1)
  t_asciidata 225*( 1) 228 ( 1)
  t_hexbstring 124*( 1) 165 ( 1) 231 ( 1)
  t_labelbuffer 123*( 1) 131 ( 1) 219 ( 1)
  t_nibble   168*( 1) 171 ( 1) 172 ( 1) 176 ( 1)
  t_phrase   121*( 1) 152 ( 1)
  t_ubyte    122*( 1) 123 ( 1) 165 ( 1) 232 ( 1)

-U-
  user_is_done 97*( 1) 104=( 1)
  user_stop    133*( 1) 307=( 1) 355=( 1) 361 ( 1)

-V-
  volume_name  96*( 1) 102 ( 1) 104 ( 1) 114*( 1) 291 ( 1) 296 ( 1) 329 ( 1)

-W-
  WRITE       101 ( 1) 259 ( 1) 266 ( 1) 272 ( 1) 301 ( 1)
  WRITELN     63 ( 1) 64 ( 1) 65 ( 1) 66 ( 1) 67 ( 1) 68 ( 1) 69 ( 1)
              70 ( 1) 71 ( 1) 84 ( 1) 85 ( 1) 100 ( 1) 155 ( 1) 237 ( 1)
              238 ( 1) 239 ( 1) 241 ( 1) 249 ( 1) 250 ( 1) 251 ( 1) 282 ( 1)

*** END Xref: 71 id's 242 references [435676 bytes/4928 id's/44603 refs]

```

File - UTIL/FILELABELS.PROCS.TEXT

< 16-DEC-1987 20:36:51 > Page - 0001

PROCEDURE/FUNCTION names FOR -LOWER-UTIL/FileLabels.TEXT

25	25	0	Lisa File_Labels	[(blank)]	-LOWER-UTIL/FileLabels.TEXT
60	60	1	Alpha		
81	81	1	Omega		
96	96	1	FetchVolumeName		
114	114	1	ExtractFileLabels		
142	142	2	ABORTFLAG		
152	152	2	ShowError		
165	165	2	byte_to_hex		
176	176	3	nibble_to_hex		
217	217	2	DumpLabelData		

\*\*\* END ProcNames: 10 PROCEDURES AND FUNCTIONS

Lisa WorkShop Utility : FileLabels [David Craig]