QL Desktop [QDT] _ QDT Icons

Version History:

V0.10	08-11-03	Split from original QDT Documentation
V0.11	09-07-03	Fixed byte offsets in IconLib_QDT format description
Fixed date for original		
V0.12	12-17-04	Removed QL color modes from spec
		Removed the confidential note
		Matched document style to User Manuals
V0.13	12-18-04	Added note about Photoshop ColorSwatch and importing

Note: Changes in text will be in this shade of red.

WebSite: http://www.jdh-stech.com

Contents

QDT Icons Overview

In QDT, icons are used to graphically display objects such as files, programs, and folders. They usually have 1-3 lines of text associated with them.

In deciding how to handle Icons, considerations were taken for disk space of each icon, disk and memory space to hold all active icons (Icon_Lib), the speed of drawing icons, and the amount of code that would be necessary to support all icon functions.

Ultimately, the decision was made to do a combination of an internal QDT sprite 256 color palette for size advantage along with the newer SMSQ/E sprite drawing capabilities and speed. See the Concepts section for details.

Icon Utilities/Functions

There are several utilities in QDT for Icon viewing and management. Each is listed here with a short description.

IconDraw_exe	Desktop ICON drawer
DefIconMan_exe	Desktop default ICON manager
IconBrowser_exe	Desktop ICON browser

Icon handling is done with common code within QDT. In the case of IconDraw which can be run outside of QDT, the needed common code is duplicated within the program. All other QDT programs call shared functions which are guaranteed to be available while QDT is running. These functions include:

```
aDrawIcon(chanid_t chan, long FileID, ColorMode, IconSize, xpos, ypos)  : changing…
short IconModes2IconFormat(short IconMode);
short IconModes(short FileType, long ptrIcon, FILE *pFile);
short IconIncluded(short IconTypeList, short IconType);
short SetToIcon(short FileType, short IconType, long ptrIcon, FILE *pFile);
short DrawIcon(short IconType, FILE *pFile, chanid_t chan, int xpos, int ypos, int scale);
```

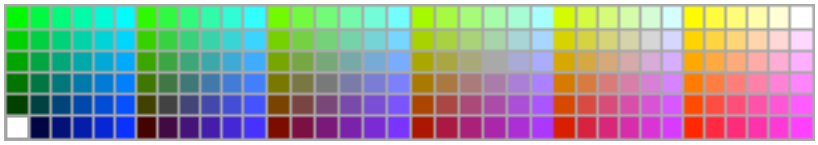
Concepts

Due to space requirements, only 256 colors at the most are supported. This means that icons can be stored in a space at the most that takes one byte per pixel, instead of 3 + bytes per pixel that ‘full’ color would require (1 byte per color per pixel = 13 x 1 + 1 allignment byte = 4 bytes).

The usage of 256 colors, if evenly spread through the visible spectrum, gives some very pleasant and professional looking icons. The need of more colors isn’t really needed for objects that are at the most only 40x40 pixels in size and not meant to be photographic image quality.

Originally, for the 256 color modes, the default SMSQ/E hi-color palette was choosen. Unfortunately, due to other requirements, this palette does not have the spread of colors needed for the icons. Instead, an internal QDT Icon palette was put together composing of what is commonly called 'Internet Safe' colors, the Netscape 216 color cube (**Error! Reference source not found.**). A good reference websight is http://the-light.com/netcol.html.

Figure 1 : Netscape 216 Color Cube



In order to take full advantage of the SMSQ/E sprite capabilities to draw the icons but still retain the smaller 256 color footprint, an internal format is used and translated live to the standard SMSQ/E full color sprite footprint when drawing each icon. This has the advantage of only a small loss of speed, the much smaller footprint of 1 byte per pixel (2 bytes per pixel if the new alpha channel is used), and much better multiple SMSQ/E cross platform support.

There can be up to 255 colors in the QDT Icon palette plus one transparency value. The QDT Icon Palette is made up of the following grouping (Table 1). Full color values, palette offset, and QDT Icon Palette Color names can be found in 000Appendix A : _*QDTsprite*_palette

Palette Entry	***Description***
0-5	Standard QL Primary Colors blue, cyan, green, yellow, red, magenta
6-32	Black, 25 grey shades, and white – dark to light . these include non-internet shade versions . internet safe entries are 6,12,17,22,27,32
33-248	Internet Safe Colors . these include the primaries, black, white, internet safe greys and colors
249-254	Not used at this time
255	Transparent

Table 1 : QDT Icon Palette Description

Importing from External Sources

IconDraw will eventually be able to import from different sources. One caution is that color translation to the QDT Icon color palette may cause some unexpected and not always desirable results. When ever possible the original should be designed in Web Safe colors (see above) or with the actual QDT Icon palette.

For the graphics designer's convenience, an Adobe Photoshop swatch is available for download through the QDT website. This will give all the colors available for the icons so that there will be no translation issues.

The final color available in the swatch is a non-display color and will translate to Transparent. for non-Alpha icons during import. Just use it to color in any area that should be transparent and IconDraw will import it as such.

Formats that IconDraw will support will be added to the IconDraw User's Manual as soon as they are implemented.

In the meantime manual conversion is available through JDH Software Technologies, using the Windows BMP format 24 color, non-compressed and non-inverted. Once the importation is enabled in IconDraw, this service will be discontinued.

Icon Management

General

The concept is to handle all Icon management procedures both live as necessary and in the background where possible. The live handling will occur when a user adds or changes an icon associated with an object. The background work will occur when the Desktop (or folder) is loaded or shutdown, and when an object's Configuration Notebook is closed and the associated icon has been changed.

QDT Desktop/Folder Startup

When the Desktop is opened, the IconLib_QDT file is checked to make sure it contains all ICONs referred to in Desktop_QDT file (all the icons needed for the desktop). If they do not exist in IconLib_QDT at that time, the library file will be rebuilt to include them. Only the current size and color format required will be loaded.

An icon pointer list is maintained with the icon location number (0-32767; -1 is the end of the list) and a pointer into the IconLib_QDT file for each icon needed by the desktop. Icons are not loaded into memory separately. Instead the IconLib_QDT file is kept open for reading only and is randomly accessed through the location pointers. QDT depends on the QDOS caching mechanism to speed up access, giving built in virtual memory management.

When the desktop is closed, any icons not included in the Desktop_QDT file are removed from the IconLib_QDT file. No other 'garbage collection on this file is maintained with the exception of an option for user to request an resync of the Icon Library from within the desktop.

Changing/Adding Icons

When a user effects an icon choice, the icon manager part of the QDT desktop program will manage the loaded Icon Libraries live. The user will only be able to effect the Icon Library directly from object configuration notebook pages where the user can modify the chosen icon. The following lists the possible events.

- New object – no icon selected
- New object – icon selected
- Changing an object – changing an icon
- Deleting an object – deletion of an icon

When creating a new object, if the user does not choose an icon, then a default icon for that type of object will be selected automatically.

If the user wants to specifically choose an icon, they are given the choice of choosing one from the loaded icons or to load one manually from a file. This is done within the configuration notebook. If the user's system has enough performance, they can elect to display the icon. Otherwise, the selection will be made by icon name only.

Once an icon has been selected (or defaulted to), the object will set its link to the icon through the Icon List. If the icon does not exist in the IconLib_QDT file, it will be loaded with the currently used mode and the Icon List will be updated.

If an object is deleted from a specific object and not used elsewhere, upon close of the desktop or resync command from the user, the icon will be removed from the IconLib_QDT file. It will not be removed during normal desktop operation.

What If

There are several times where the icon just can't be displayed as available, etc. If the system can not find the icon of the correct size, it will then display the system default for that type of object.

File Formats

IconName_icon

Icons can also be stored independently of the IconLib_QDT file which helps in managing and trading icons separately from the desktop. These are stored in a similar format to that of the IconLib_QDT file but without the linked list information. Each file can contain 1 to 2 sizes allowed [only 256 color mode is now allowed]. The following is the format icons are saved in. The different formats are always saved in the ordered sequence as shown in the Formats Byte (starting with format 1, then going to format 2 – 4 – 8).

V0.11 Format

4 Modes Format [QL and 256 colors] – no longer supported, see next page for current format

Bytes	Info	
0 : 6	"QDTICON"	file header
7 : 8	Reserved	included so Formats Byte is same offset in all icon file formats
9	Formats Included	bit set = included, clear = not included
	bit # 0	4 color:Small
	bit # 1	4 color:Normal
	bit # 2	256 color:Small
	bit # 3	256 color:Normal
	set = yes, clear = no	
	bit # 4	*use alpha instead of a mask*
	bit # 5	*auto alpha blend over transparent (not implemented yet)*
	bit # 6	*RLE encode alpha mask*
	bit # 7	*RLE encoding primary sprite*
	Offset in file from byte 0 to each sprite type (format byte), 0 = not available	
10-11	4 color:Small	offset = 34
12-13	4 color:Normal	offset = 34 + 1600 + 1
14-15	256 color:Small	offset = 34 + 1600 + 1 + 400 + 1
16-17	256 color:Normal	offset = 34 + 1600 + 1 + 400 + 1 + 1600 + 1
18 : 33	Icon Name	(15 characters max length, 0 terminated, 0 fill)
34	Icon format	0 = no more date (file will always be one byte longer than data)
…	Icon data	stored as rows of data as stored in screen memory
		length and row count is dependant on format
...	Icon format	0 = no more data
…	or repeat data/format unless no more icon formats stored	

V0.12 Format

255 Color Only Format – currently supported

Bytes	Info	
0 : 6	"QDTICON"	file header
7 : 8	Reserved	included so Formats Byte is same offset in all icon file formats
9	Formats Included	bit set = included, clear = not included
	bit # 0	reserved, set to 0
	bit # 1	reserved, set to 0
	bit # 2	256 color:Small
	bit # 3	256 color:Normal
	set = yes, clear = no	
	bit # 4	*use alpha instead of a mask*
	bit # 5	*auto alpha blend over transparent (not implemented yet)*
	bit # 6	*RLE encode alpha mask*
	bit # 7	*RLE encoding primary sprite*
	Offset in file from byte 0 to each sprite type (format byte), 0 = not available	
10-11	256 color:Small	offset = 34
12-13	256 color:Normal	offset = 34 + 1600 + 1
14 : 29	Icon Name	(15 characters max length, 0 terminated, 0 fill)
30	Icon format	0 = no more date (file will always be one byte longer than data)
…	Icon data	stored as rows of data as stored in screen memory
		length and row count is dependant on format
...	Icon format	0 = no more data
…	or repeat data/format unless no more icon formats stored	

Notes:
- if alpha mask choosen, only available in 256 color mode (4 color mode stays with transparency only)
- alpha mask data follows exactly at the end of the color bit normal data if it exists

IconLib_QDT

This file holds the images of all the icons used by the desktop. It can be considered as an icon library. Icons stored in it can be designed with the QDTIconDraw_exe utility or imported from several popular ICON formats with the same utility.

The QL Desktop comes shipped with several default icons. These icons can not be deleted (but they can be updated). If the desktop calls for an icon which it can't find in the icon library file, it will use one of the following default icons instead.

Folder : Open	Folder : Closed
Program	
Text	
Graphic	
Unknown	

Each icon is stored in one of four different formats.

4 color:Normal	4 color:Small
256 color:Normal	256 color:Small

Normally an icon only has one format in use during a desktop session. However, multiple formats are allowed and are stored as separate entries in the IconLib file.

Looking at the entries for the individual icons within the IconLib file, the bytes follow closely the original individual icon file format. Bytes 8 and 9 are identical (default and format information). Note that byte 9 shows all the formats originally available in the individual icon file, not just the one included in this entry. This is to allow the user to quickly see options when picking icons sizes, even if it isn't loaded (avoid having to go find and open the original source file each time to check).

The following 8 bytes in the individual icon file are skipped in the IconLib file as there is only one format per Icon entry in this library and pointer information is therefore not needed. Instead, after the formats byte, the Icon Name follows immediately and the individual Icon Format byte reflects the single format for this entry (unlike byte 9 which showed all available formats).

Format

Binary : Linked List : File Header

File Header Format :

Bytes	Info	
10 : 0	"ICONLIB_QDT"	file header
11	compressed	0 = no, 1 = yes (compression uses RLE encoding)
15:12	reserved, set to 0	

First entry :

Entry Format :

3:0	Pointer to next entry (-1 if last)	
7:4	Pointer to previous entry (-1 if 1st)	
8	Default	0 = no, 1 = yes (1 = replaceable, but not deletable)
9	Formats Included	bit set = included, clear = not included

	bit # 0	reserved – set to 0
	bit # 1	reserved – set to 0
	bit # 2	256 color:Small
	bit # 3	256 color:Normal
	set = yes, clear = no	
	bit # 4	*use alpha instead of a mask*
	bit # 5	*auto alpha blend over transparent (not implemented yet)*
	bit # 6	*RLE encode alpha mask*
	bit # 7	*RLE encoding primary sprite*
10:25	Icon Name	(15 characters max length, 0 terminated, 0 fill)
26	Icon format	0 = no more date (file will always be one byte longer than data)
…	Icon data	stored as rows of data as stored in screen memory length and row count is dependant on format
...	Icon format	0 = no more data
…	or repeat data/format unless no more icon formats stored	

The icon data is stored in horizontal lines starting from the lower left hand corner of the icon. The RGB format uses 256 color palette entries so it is stored in a byte per pixel. Mode 4 also requires one byte per pixel and follows the standard QL 4 color mode (0-7).

Appendix A : **_QDTsprite*_palette***

This is the list of QDT Icon Sprite Palette values, palette locations and QDT specific color names. It is taken directly from the assembly code file ‘spritepalette_s’.

```
  .text
  .even

  .globl _QDTsprite_palette
;
_QDTsprite_palette:
primary:
  dc.l  $0000FF00   ; blue        0
  dc.l  $00FFFF00   ; cyan        1
  dc.l  $00FF0000   ; green       2
  dc.l  $FFFF0000   ; yellow      3
  dc.l  $FF000000   ; red         4
  dc.l  $FF00FF00   ; magenta     5

grey:
  dc.l  $00000000   ; black       6
  dc.l  $09090900   ; grey25      7
  dc.l  $11111100   ; grey24      8
  dc.l  $19191900   ; grey23      9
  dc.l  $21212100   ; grey22      10
  dc.l  $29292900   ; grey21      11
  dc.l  $33333300   ; grey20      12
  dc.l  $3D3D3D00   ; grey19      13
  dc.l  $47474700   ; grey18      14
  dc.l  $51515100   ; grey17      15
  dc.l  $5B5B5B00   ; grey16      16
  dc.l  $66666600   ; grey15      17
  dc.l  $70707000   ; grey14      18
  dc.l  $7A7A7A00   ; grey13      19
  dc.l  $84848400   ; grey12      20
  dc.l  $8E8E8E00   ; grey11      21
  dc.l  $99999900   ; grey10      22
  dc.l  $A3A3A300   ; grey9       23
  dc.l  $ADADAD00   ; grey8       24
  dc.l  $B7B7B700   ; grey7       25
  dc.l  $C1C1C100   ; grey6       26
  dc.l  $CCCCCC00   ; grey5       27
  dc.l  $D6D6D600   ; grey4       28
  dc.l  $E0E0E000   ; grey3       29
  dc.l  $EAEAEA00   ; grey2       30
  dc.l  $F4F4F400   ; grey1       31
  dc.l  $FFFFFF00   ; white       32
```

```
color:
  dc.l  $00000000             ;   33
  dc.l  $00003300             ;   34
  dc.l  $00006600             ;   35
  dc.l  $00009900             ;   36
  dc.l  $0000CC00             ;   37
  dc.l  $0000FF00             ;   38
  dc.l  $00330000             ;   39
  dc.l  $00333300             ;   40
  dc.l  $00336600             ;   41
  dc.l  $00339900             ;   42
  dc.l  $0033CC00             ;   43
  dc.l  $0033FF00             ;   44
  dc.l  $00660000             ;   45
  dc.l  $00663300             ;   46
  dc.l  $00666600             ;   47
  dc.l  $00669900             ;   48
  dc.l  $0066CC00             ;   49
  dc.l  $0066FF00             ;   50
  dc.l  $00990000             ;   51
  dc.l  $00993300             ;   52
  dc.l  $00996600             ;   53
  dc.l  $00999900             ;   54
  dc.l  $0099CC00             ;   55
  dc.l  $0099FF00             ;   56
  dc.l  $00CC0000             ;   57
  dc.l  $00CC3300             ;   58
  dc.l  $00CC6600             ;   59
  dc.l  $00CC9900             ;   60
  dc.l  $00CCCC00             ;   61
  dc.l  $00CCFF00             ;   62
  dc.l  $00FF0000             ;   63
  dc.l  $00FF3300             ;   64
  dc.l  $00FF6600             ;   65
  dc.l  $00FF9900             ;   66
  dc.l  $00FFCC00             ;   67
  dc.l  $00FFFF00             ;   68

  dc.l  $33000000             ;   69
  dc.l  $33003300             ;   70
  dc.l  $33006600             ;   71
  dc.l  $33009900             ;   72
  dc.l  $3300CC00             ;   73
  dc.l  $3300FF00             ;   74
  dc.l  $33330000             ;   75
  dc.l  $33333300             ;   76
  dc.l  $33336600             ;   77
```

```
dc.l  $33339900             ;   78
dc.l  $3333CC00             ;   79
dc.l  $3333FF00             ;   80
dc.l  $33660000             ;   81
dc.l  $33663300             ;   82
dc.l  $33666600             ;   83
dc.l  $33669900             ;   84
dc.l  $3366CC00             ;   85
dc.l  $3366FF00             ;   86
dc.l  $33990000             ;   87
dc.l  $33993300             ;   88
dc.l  $33996600             ;   89
dc.l  $33999900             ;   90
dc.l  $3399CC00             ;   91
dc.l  $3399FF00             ;   92
dc.l  $33CC0000             ;   93
dc.l  $33CC3300             ;   94
dc.l  $33CC6600             ;   95
dc.l  $33CC9900             ;   96
dc.l  $33CCCC00             ;   97
dc.l  $33CCFF00             ;   98
dc.l  $33FF0000             ;   99
dc.l  $33FF3300             ;  100
dc.l  $33FF6600             ;  101
dc.l  $33FF9900             ;  102
dc.l  $33FFCC00             ;  103
dc.l  $33FFFF00             ;  104

dc.l  $66000000             ;  105
dc.l  $66003300             ;  106
dc.l  $66006600             ;  107
dc.l  $66009900             ;  108
dc.l  $6600CC00             ;  109
dc.l  $6600FF00             ;  110
dc.l  $66330000             ;  111
dc.l  $66333300             ;  112
dc.l  $66336600             ;  113
dc.l  $66339900             ;  114
dc.l  $6633CC00             ;  115
dc.l  $6633FF00             ;  116
dc.l  $66660000             ;  117
dc.l  $66663300             ;  118
dc.l  $66666600             ;  119
dc.l  $66669900             ;  120
dc.l  $6666CC00             ;  121
dc.l  $6666FF00             ;  122
dc.l  $66990000             ;  123
dc.l  $66993300             ;  124
```

```
dc.l  $66996600             ;  125
dc.l  $66999900             ;  126
dc.l  $6699CC00             ;  127
dc.l  $6699FF00             ;  128
dc.l  $66CC0000             ;  129
dc.l  $66CC3300             ;  130
dc.l  $66CC6600             ;  131
dc.l  $66CC9900             ;  132
dc.l  $66CCCC00             ;  133
dc.l  $66CCFF00             ;  134
dc.l  $66FF0000             ;  135
dc.l  $66FF3300             ;  136
dc.l  $66FF6600             ;  137
dc.l  $66FF9900             ;  138
dc.l  $66FFCC00             ;  139
dc.l  $66FFFF00             ;  140

dc.l  $99000000             ;  141
dc.l  $99003300             ;  142
dc.l  $99006600             ;  143
dc.l  $99009900             ;  144
dc.l  $9900CC00             ;  145
dc.l  $9900FF00             ;  146
dc.l  $99330000             ;  147
dc.l  $99333300             ;  148
dc.l  $99336600             ;  149
dc.l  $99339900             ;  150
dc.l  $9933CC00             ;  151
dc.l  $9933FF00             ;  152
dc.l  $99660000             ;  153
dc.l  $99663300             ;  154
dc.l  $99666600             ;  155
dc.l  $99669900             ;  156
dc.l  $9966CC00             ;  157
dc.l  $9966FF00             ;  158
dc.l  $99990000             ;  159
dc.l  $99993300             ;  160
dc.l  $99996600             ;  161
dc.l  $99999900             ;  162
dc.l  $9999CC00             ;  163
dc.l  $9999FF00             ;  164
dc.l  $99CC0000             ;  165
dc.l  $99CC3300             ;  166
dc.l  $99CC6600             ;  167
dc.l  $99CC9900             ;  168
dc.l  $99CCCC00             ;  169
dc.l  $99CCFF00             ;  170
dc.l  $99FF0000             ;  171
```

```
dc.l  $99FF3300             ;  172
dc.l  $99FF6600             ;  173
dc.l  $99FF9900             ;  174
dc.l  $99FFCC00             ;  175
dc.l  $99FFFF00             ;  176

dc.l  $CC000000             ;  177
dc.l  $CC003300             ;  178
dc.l  $CC006600             ;  179
dc.l  $CC009900             ;  180
dc.l  $CC00CC00             ;  181
dc.l  $CC00FF00             ;  182
dc.l  $CC330000             ;  183
dc.l  $CC333300             ;  184
dc.l  $CC336600             ;  185
dc.l  $CC339900             ;  186
dc.l  $CC33CC00             ;  187
dc.l  $CC33FF00             ;  188
dc.l  $CC660000             ;  189
dc.l  $CC663300             ;  190
dc.l  $CC666600             ;  191
dc.l  $CC669900             ;  192
dc.l  $CC66CC00             ;  193
dc.l  $CC66FF00             ;  194
dc.l  $CC990000             ;  195
dc.l  $CC993300             ;  196
dc.l  $CC996600             ;  197
dc.l  $CC999900             ;  198
dc.l  $CC99CC00             ;  199
dc.l  $CC99FF00             ;  200
dc.l  $CCCC0000             ;  201
dc.l  $CCCC3300             ;  202
dc.l  $CCCC6600             ;  203
dc.l  $CCCC9900             ;  204
dc.l  $CCCCCC00             ;  205
dc.l  $CCCCFF00             ;  206
dc.l  $CCFF0000             ;  207
dc.l  $CCFF3300             ;  208
dc.l  $CCFF6600             ;  209
dc.l  $CCFF9900             ;  210
dc.l  $CCFFCC00             ;  211
dc.l  $CCFFFF00             ;  212

dc.l  $FF000000             ;  213
dc.l  $FF003300             ;  214
dc.l  $FF006600             ;  215
dc.l  $FF009900             ;  216
dc.l  $FF00CC00             ;  217
```

```
dc.l  $FF00FF00             ;  218
dc.l  $FF330000             ;  219
dc.l  $FF333300             ;  220
dc.l  $FF336600             ;  221
dc.l  $FF339900             ;  222
dc.l  $FF33CC00             ;  223
dc.l  $FF33FF00             ;  224
dc.l  $FF660000             ;  225
dc.l  $FF663300             ;  226
dc.l  $FF666600             ;  227
dc.l  $FF669900             ;  228
dc.l  $FF66CC00             ;  229
dc.l  $FF66FF00             ;  230
dc.l  $FF990000             ;  231
dc.l  $FF993300             ;  232
dc.l  $FF996600             ;  233
dc.l  $FF999900             ;  234
dc.l  $FF99CC00             ;  235
dc.l  $FF99FF00             ;  236
dc.l  $FFCC0000             ;  237
dc.l  $FFCC3300             ;  238
dc.l  $FFCC6600             ;  239
dc.l  $FFCC9900             ;  240
dc.l  $FFCCCC00             ;  241
dc.l  $FFCCFF00             ;  242
dc.l  $FFFF0000             ;  243
dc.l  $FFFF3300             ;  244
dc.l  $FFFF6600             ;  245
dc.l  $FFFF9900             ;  246
dc.l  $FFFFCC00             ;  247
dc.l  $FFFFFF00             ;  248
dc.l  $FFFFFF00             ;  249
dc.l  $FFFFFF00             ;  250
dc.l  $FFFFFF00             ;  251
dc.l  $FFFFFF00             ;  252
dc.l  $FFFFFF00             ;  253
dc.l  $FFFFFF00             ;  254
dc.l  $FFFFFF00             ;  255
```