

QDT Notebook User Manual

Version History:

01-23-05
01-19-14

Vers 1.00
Vers 1.02

Original
Updated web links, minor cleanups
Freeware Version

Copywrite ©:

2005, 2014

James Hunkins [JDH Software Technologies]

Website:

<http://www.jdh-stech.com>

Contents

Contents	1
Introduction	3
Basic Information.....	3
Section Description	3
Saving Changes	3
Loose Items and Exiting	4
Basic Controls.....	4
Tab Arrows	4
Short Cut Keys	4
Default Entry.....	5
Buttons	5
Text Entry Boxes	5
CheckBox	6
ExclusiveList Buttons	6
Scrolling Menus.....	6
Notebooks.....	7
Common Pages	7
IconOpen / IconClosed / IconExec.....	7
Global Variables	8
Desktop.....	8
Object	8
IconOpen	9
StartUp	9
Archive	10
Display	12
ScreenSaver	15
QDOptions	15
SysOptions	16

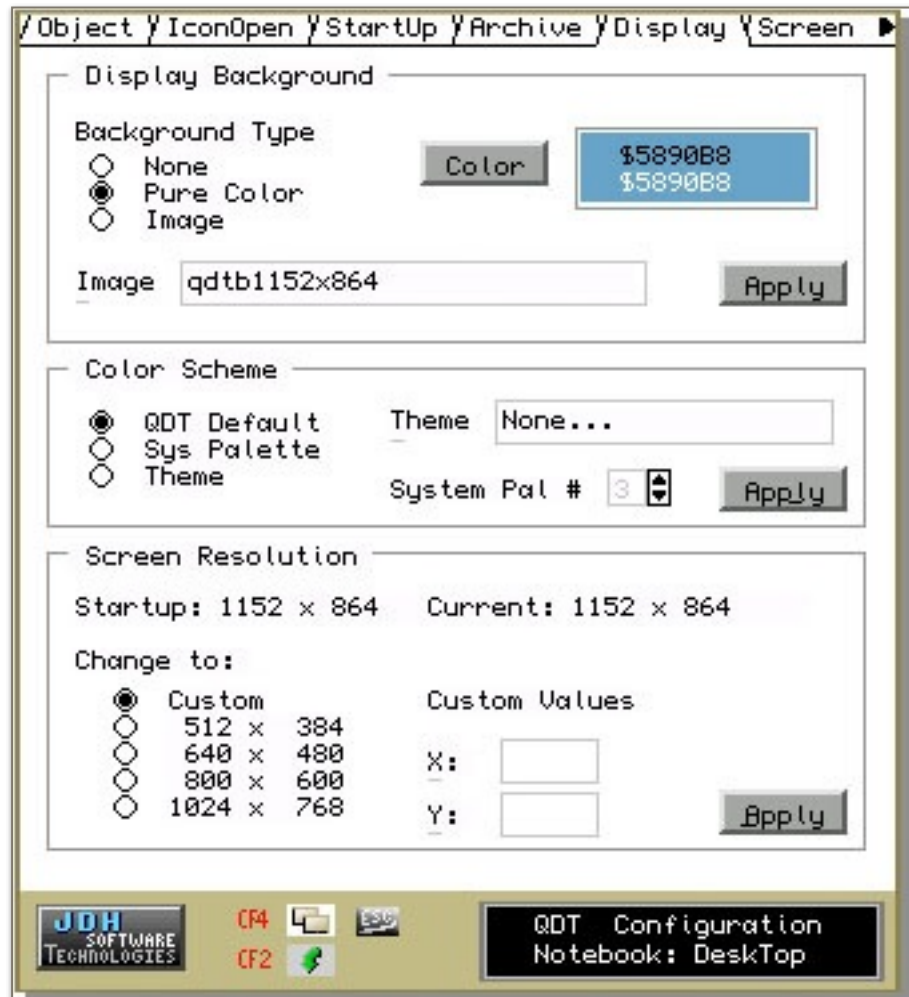
MenuItems	16
HardwareInfo	16
SysSoftwareInfo	16
Executable File	17
Object	17
Session	18
Associations	18
ConfigBlkProp	19
IndFileProp	19
Folder – Normal and Executable	19
Object	19
Session [exec only]	20
Associations [exec only]	21
MenuItems	21
ConfigBlkProp [exec only]	21
IndFileProp [exec only]	21
File – Text, Graphics, and Unknown	21
Object	22
Session	22
Associations	22
Printer	23
File Manager	23
Thing	23
Appendixes	24
<i>A: Software License and Warranty</i>	<i>24</i>
<i>B: Notebook Entries : by name and page</i>	<i>27</i>
<i>C: QCP Usage Notes</i>	<i>31</i>
<i>D: Capturing a Background Image for use in QDT</i>	<i>32</i>

Introduction

The QDT Tabbed Notebook was developed to allow setting and configuration for a wide variety of items. Each type of object has its own specific notebook with multiple tabbed pages. Each page presents information and/or user configurable settings.

A notebook is reached from an object's drop-down menu 'Property' selection. In the case of the QDT Desktop notebook, along with the QDT desktop object's menu, the drop-down menu can also be reached from the QDT hotkey drop-down menu.

This manual starts with basic information on how a notebook is structured, then the different controls, and finally it steps through each individual notebook type and each page and item.



In the Appendix sections, there are a listing of each user settable command, arranged by notebook/page. This will help the user do quick look ups and find the information that they need quickly.

Basic Information

Section Description

A notebook display is made up of three parts. The bottom holds the standard QL loose items Move, Refresh and ESC, along with which notebook it is.

The center part of the window is the primary control window. It shows a page of settings at a time and information that often can be changed by the user. This is where most of the activity occurs.

The top part of the window is a series of tabs. Since a notebook consists of multiple pages, changing to a new page is as simple as clicking on the tab for the desired page.

Saving Changes

As discussed in the QDT User Manual and elsewhere, changing a desktop is done in two stages. The current active desktop gets any changes as soon as they are made unless otherwise specified. However, these changes to a desktop's settings are not physically saved to the QDT

database on the drive until the user specifies for it to happen. This provides a convenient method to through away accidental changes or changes made for just a single session.

The notebook also has a few rules as to how it saves things to the current active desktop. In most cases, as long as the user does not change a page, nothing is saved. Only by changing to a different notebook page or closing the notebook are settings updated to the active desktop. This allows the user to correct any non-intended change before continuing.

For example, if a user sets an incorrect background format in the QDT Desktop notebook, they can fix this before it is saved to the desktop when they change pages or apply the change directly.

This example brings up the second type of saving of a change. Some changes will only affect the desktop either on the restart of QDT or, in many cases, when the corresponding 'Apply' button is pressed. These conditions apply to changes that could have more 'dramatic' impact on the desktop, such as changing a background image or changing screen resolution.

Loose Items and Exiting

A notebook presents three standard loose items as given here with their short cut keys; Move [CF4], Refresh [CF2], and ESC [ESC]. The operations are standard QL.

To quit the notebook, pressing the ESC loose item or hitting the 'ESC' key closes the notebook and saves any changes from the last displayed page to the currently active desktop.

Basic Controls

There are quite a few different controls to make selection of a wide variety of options easier. The following sections cover each of these and all their special cases.

Most notebook responses can be done directly with the mouse. Use of the left or right mouse buttons will be referred to as 'LMB' – Left Mouse Button, and 'RMB' – Right Mouse Button. For long time QL users, the LMB is often referred to as 'hit' and can also be done by using the 'SPACE' bar. The RMB is often referred to as an 'enter' and can be accomplished with the 'RETURN' key.

Tab Arrows

If a notebook has more pages than it can display the tabs for at one time, small arrows will appear on either side of the tabs along the top of the window. This indicates that there are more tabs available in the designated direction.



Figure 2 : Arrow on tab bar

To access these hidden tabs, simply click on the arrow and the tabs will scroll over.

The page will not actually change until the user clicks specifically on the desired tab.

Short Cut Keys

Many of the entry options can be accessed through short cut keys. If the item has an underlined letter in it, that is the short cut key. For example, in Figure 4 in the Buttons section, 'R' is the short cut key for the 'Restart' button.

As long as the cursor is located anywhere in the main window area, these shortcut keys will give the same action as clicking on the object directly. The notebook will not respond to them if the cursor is outside this area.

Some controls have different responses depending on whether a left or right mouse click is taken. The short cut keys will always be equivalent to a RMB (Right Mouse Button).

Default Entry

Some items on different pages may be filled in with a system or QDT default value if the user has not previously entered a value for that item. If this is the case a small, slanted 'sd' with a squiggle under it is displayed next to the entry field. Once the entry has been changed by the user, this symbol will disappear.

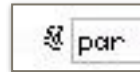


Figure 3 : Default Entry Symbol

Buttons

Buttons are simply that, buttons. By pushing/clicking on them, an action is caused to occur. The text on or near to the button should explain what will happen.

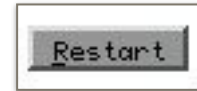


Figure 4 : Button Control

Some buttons will cause something to be applied or changed. For example, after picking a color for the desktop background, there is an 'Apply' button which causes the desktop's background to change to that color.

Other buttons call up external programs to perform some function. For example the 'IconDraw' button on some pages will open the external IconDraw editing program.

There are two types of external program call ups. The first one calls the program and locks the notebook window until the program has been closed. An example of this is the 'Edit' button on the Icon pages of notebooks. The idea is that the user would be doing a specific function that needs to be finished and applied to the notebook before continuing on with the notebook (the user can always cancel from such programs). This example calls up the IconDraw editor with the icon shown in the notebook's window for modification.

The other external call type is one that starts a program but doesn't need to wait for it to finish. The 'IconDraw' example just given is one of these. Unlike the 'Edit' button example, this just opens IconDraw program for general use as a convenience and is not tied in anyway to the notebook. Therefore the notebook doesn't have to wait for the program to close.

Text Entry Boxes

Text entry boxes allow the user to enter text directly. They are commonly used to setup labels such as object text for the desktop or to enter file names and directories.



Figure 5 : Text Entry Box examples

Some text entry boxes may have text in light grey versus the normal darker text color. These are not changeable by the user and are displayed just to provide information. Trying to click on them will give no results.

Text entry boxes do have different responses in most cases depending on whether the LMB or RMB (or short cut key) is used.

By using the LMB, text is directly entered into the text field by the user with many of the common QL editing keystrokes. The edit mode is always 'overwrite' versus 'insert'. The key strokes available, in addition to normal letter keys, are:

Table 1 : Text Editing Keys

Left	move cursor left one character
Right	move cursor right one character
Alt Left	move cursor to the first character on the left
Alt Right	move cursor to the last character on the right
Delete	deletes character under the cursor
Back Space	deletes character to the left of the cursor
Shift Left	moves cursor to first non ASCII character to the left such as an '-' or '_'
Shift Right	moves cursor to first non ASCII character to the right such as an '-' or '_'
Enter	exits edit mode and saves the edited results
ESC	exits edit mode and restores the original entry, discarding any edits done

For text entries that may be displayed as multiple lines (see the notebook page information following), a line break can be forced with the characters '\n'. These characters will not be displayed but instead will force the next character to start on a new line.

The RMB in many cases (not all, a few entries will only allow pure text entry) takes advantage of the QMenu extensions and generates an external window to allow the user to directly make a selection which is copied conveniently back into the text entry box.

There are three types of external windows that can occur.

- 1) File name : this will bring up the QMenu File menu window for normal selection of a file name. In some cases, where a required suffix exists, the filename suffix will be set in the menu for the user, limiting the displayed files to the proper type.
- 2) Directory name : this will bring up the QMenu Directory menu window for normal selection.
- 3) List of files or other objects : QDT has several cases where types of files must exist in one directory only and with a common suffix. For the user's convenience, these entries will generate a list of the available files in a simple QMenu list window.

CheckBox

Checkboxes are shaped as squares. Clicking on one toggles between checked (an 'x') or unchecked (empty box). Checkboxes are used to set or unset a feature.

Figure 6 :
check
boxes



ExclusiveList Buttons

Exclusive lists are lists of items of which only one can be selected at a time. They are distinguished as being a vertical grouping of items with circles next to each item.

When selecting an item (a circle with a solid dot in the middle), if another was previously selected it gets automatically unselected (an empty circle) as the new one is selected.

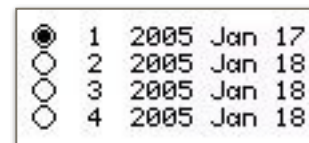
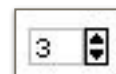


Figure 7 :
Exclusive list

Scrolling Menus

A scrolling menu is a list of items of which only one item is visible at a time.

Figure 8 :
Scrolling
Menu



It will always have an up and down arrow to the side of the list item displayed.

Scrolling the list is done by either clicking on the up or down arrow or by clicking on the list item itself and using the 'Up' or 'Down' keys on the keyboard.

The item within the scrolling menu box is not directly editable by the user.

Notebooks

The following pages cover each notebook, page by page. Descriptions of each item per page are given with specific implementation notes as needed.

Common Pages

IconOpen / IconClosed / IconExec

These pages are available in all the standard notebooks. They control the object's appearance and text label. All notebooks except the QDT Desktop have two of these pages, including the IconOpen page. Executable programs have the IconExec page while the other notebooks will show the IconClosed page. Since the desktop only has one state (it is always open), it only shows the IconOpen page.

Information Displayed

Icon Image

This is the current image of the icon that will be displayed on the desktop or in the folder for this object. If the icon name or icon size is not available, then the QDT default icon image for the type of object will be displayed instead.

Sizes Available

The sizes that are available in the icon file selected have a green check mark displayed. Choosing a size that is not checked will force QDT to display the alternate system default icon for the object instead.

Current Size

This is the size of the icon that will be displayed.

Notes

This area displays any relevant notes for this object's entry.

User Entries

Icon Text

{ text entry box – max length = 3 lines of 16 characters, 48 chars }

This is the actual text displayed under an objects icon either on the desktop or in a folder. '\n' in this entry will not be displayed on the desktop or within a folder. Instead they will force a line return at that point in the text.

Icon Name

{ text entry box – max length = 25 chars }

This is the name of the icon file without the _ICON suffix. All icons used must be in the QDT location specified in the environmental variable 'QDTICON_LOC'. The directory information is not included in this entry.

RMB will bring up a list of all available icons. LMB allows manual text entry.

Size : Small, Normal

{ exclusive list }

This selects either a 20x20 (small) or a 40x40 (normal icon size).

Edit

{ button – program, waits for program to close }

This brings up the IconDraw Icon Editor with the icon for this object loaded into it. The notebook will freeze until this copy of IconDraw is closed.

Here the user can modify the look of the icon chosen. However, the updated image will not display until QDT is restarted.

IconDraw

{ button – program, does not wait for program to close }

This brings up an independent copy of IconDraw with no icon pre-loaded.

Find

{ button – program, waits for program to close }

This brings up the icon browser to allow for selecting an icon by image.

Global Variables

This page [not implemented yet] will display all environmental variables and allow user modification of each one. Note that environmental and global variables are commonly names for the same thing and can be intermixed depending on the discussion.

Some changes of the QDT variables will not take affect until restarting QDT.

Desktop

This notebook is opened from either the QDT desktop icon drop-down menu or the QDT shortcut key drop-down menu – Properties.

Object

Information Displayed

Object Name

This is the primary QDT master program. It will always be QDT_exe.

Object Location

This is the directory that the QDT master program is located in. It has the same value as the environmental variable 'QDT_BIN_LOC'.

Config Block Name

This is the program name found in the standard QJump type config block. Not all programs have Config blocks and some do not support the name feature. In these cases either a message (block not found) or possibly nonsense text will be displayed.

Config Block Vers

This is the program's version number found in the QJump type config block.

Object Type

This is the QDT object type. For this type of object, it will display Desktop Folder.

QDT Location

This is the primary directory location for QDT. It has the same value as the environmental variable 'QDT_LOC'.

QDT Graphics Location

This is the parent location for any QDT used graphics. For example, QDT icons are kept in this location plus ICONS_. QDT backgrounds are kept in this location plus BACKGNDS_. It has the same value as the environmental variable 'QDT_ICON_LOC'.

It is entirely possible with some users that they may have enough graphics and backgrounds that they may want to store them on another disk partition or different disk entirely. This would allow for that circumstance.

QDT Icon Lib Location

This is the directory location that holds the QDT_ICONLIB file. This file is the real time database of all icons currently in use by the desktop, versus the individual Icon files of all icons available to use by QDT. It has the same value as the environmental variable 'QDT_ICONLIB_LOC'.

QDT Program Location

This is the directory location that holds all QDT programs (binary files). It has the same value as the environmental variable 'QDT_BIN_LOC'.

User Entries

Output Dev

{ text entry box – max length = 44 chars }

This is the system's output device. It initially has the QL default value found in environmental variable 'SPL_USE'. It can be changed to any value desired and any QL launched program will use the new value. Once changed the 'sd' symbol next to it will be removed.

IconOpen

See the previous Common Pages : IconOpen / IconClosed / IconExec section for details.

StartUp

Information Displayed

-none-

User Entries

Desktop Startup File

{ text entry box – max length = 64 chars }

QDT allows for a SBasic type file to be loaded and run upon startup, allowing the user to automate different tasks. The file should have no line numbers. Only files in the QDT_LOC directory can be accessed. All startup files must end with the suffix _STUP.

When entering a name directly through the use of the LMB, only enter the file name and do not include the suffix.

The RMB or shortcut key brings up a list window of all available startup files.

Show Desktop Object on Desktop

{ check box }

It is possible to not display the QDT object on the desktop to save desktop space. By unselecting this checkbox, when QDT next launches its QDT object will not be displayed.

CAUTION: *it is critical that, if the QDT object is not displayed, the hotkey function is enabled (enabled by default). The QDT hotkey will be the only way to access the QDT drop-down menu if the QDT object is not displayed on the desktop.*

Use/Restore Desktop : Alternate File

{ text entry box – max length = 64 chars }

By default, QDT will always use the DESKTOP_QDT file for its database when launching. It is possible however to have QDT switch to an alternate database file. This can be one of the archived database files or one that was manually saved.

This can be useful if there are different desktop configurations for different projects or different users. By picking, for example, DESKTOP_MARY_QDT and restarting QDT, it will then replace the current DESKTOP_QDT file with the DESKTOP_MARY_QDT file and restart with the different desktop next time. Note that the original DESKTOP_QDT file is not saved other than by the standard archiving if enabled.

This text entry is only a name entry. It is necessary to choose the Change or Restart buttons to cause the files to be changed. This extra level of precaution is done as this is a critical change.

CAUTION: *if the user saves their desktop after this change, it will replace the alternate desktop with the currently displayed one which is being saved, thereby overriding this action.*

Only files that start with DESKTOP and end with _QDT and which are located in the QDT_LOC will be recognized.

The LMB allows direct text input but care must be used in following the above naming protocol. The RMB or shortcut key will bring up a list window to select from.

Use/Restore Desktop : Change

{ button – action }

This causes the file named in the 'Alternate File' field to be copied over the top of the current DESKTOP_QDT database file. Do not save the current desktop after this action before a restart in order to avoid losing the change.

Use/Restore Desktop : Restart

{ button – action }

This causes the file named in the 'Alternate File' field to be copied over the top of the current DESKTOP_QDT database file. It then restarts QDT automatically.

Save Desktop : Current

{ button – action }

This saves the current desktop and any changes that have been made. It is the same as the 'Save' in the QDT object's drop-down menu.

Save Desktop : Other

{ button – action }

This allows the saving of the current desktop and any changes into an alternate file name. It is useful for creating special purpose desktops for different projects or users. See the previous Use/Restore Desktop : Alternate File section for more details.

Theme

{ button – program, waits for program to close }

This starts the Theme Manager program. This program allows the user to change full desktop themes or individual components, such as colors, icons sets, etc.

Archive

Archiving is an important way to save copies of older desktop databases and also protecting against accidental changes/saves or file corruption.

When a desktop is archived, the archival copy is saved as DESKTOP_hhhhhhhh_QDT, where hhhhhhhh represents an eight digit hexadecimal number of the current QL time/date in the QL low level time/date format.

Archiving is on by default only for manual saves and if the desktop has changed when a QDT session is ended.

Information Displayed

-none-

User Entries

Archive Desktop

{ check box }

If checked, archiving is enabled based on the rules in the rest of this page. If not checked, there is no archiving at all. Saving a desktop will overwrite the last copy and no recovery will be available.

The following archive parts only work if this box is checked.

At Every QDT startup

{ check box }

At every start of QDT session, the current desktop gets copied to an archived version.

At Next QDT Startup Only

{ check box }

At the start of the next QDT session, the current desktop will get copied to an archived version. Then this function will be cleared.

If Desktop has Changed

{ check box }

When quitting QDT, if this is checked and the desktop has changed since the last desktop save, then the user will be prompted to see if they want to save the desktop.

Per number of days

{ check box }

If checked, upon start QDT will look to see when the last archive file was saved, compare that against the current date and how many days are set in this page. If the number of elapsed days is equal or greater than the set number of days, upon startup QDT will make an archived copy.

Number of days

{ text entry box – max length = 2 chars, numbers only }

This text entry box does not have a direct label but is found beside the 'Per number of days' check box entry.

Only numbers are allowed, up to the value of 99. Only direct text entry can be used.

If manually saving desktop

{ check box }

If checked, whenever the user saves the desktop from a drop-down menu or prompt, the previous copy of DESKTOP_QDT will be archived before the save overwrites it.

Show archive status during startup

{ check box }

If checked, upon QDT startup, if there is any archive action taken, a status window will display what is happening. The user will need to check OK in the status window to continue with the QDT startup.

Keep [number] Separate Archives

{ scrolling menu }

Between 1 and 4 archives can be kept at one time by QDT. If archiving is on and the number of old archive files is greater than the number set here, then the oldest files will be removed on the next QDT startup.

If more than 4 archives are desired, then it is necessary to move or rename to none archival naming conventions the extra archives desire.

This function will not remove any file that does not follow the naming conventions given earlier.

Archive Location

{ text entry box – max length = 64 chars }

By default, archive files are kept in the same location as the original DESKTOP_QDT database file is kept (QDT_LOC). However, with this entry it is possible to specify a different location.

If this is changed, then QDT will not recognize the archives in the old location.

Recover Archived Desktop

{ exclusive list }

This dynamically generated list will display any available archive files (four maximum). This selection, in combination with the 'Recover' button, will copy the checked archive file over the top of the current DESKTOP_QDT file.

Recover

{ button – action }

By pressing this button, the archive file checked in the Recover Archived Desktop list will be copied over the top of the current DESKTOP_QDT database file, allowing recovery of older desktop configurations.

CAUTION: *if a save is done of the desktop before a restart, it will overwrite this recovery.*

Display

This page allows the user to dynamically alter the appearance of the current display. Since these are all critical changes, the user will need to press the 'Apply' button in the respective section before a physical change in this session actually occurs.

These options will get saved to the desktop database (if the user saves it) and therefore will be active on the next QDT start, even if the 'Apply' button is not pressed.

This page can be seen in Figure 1 in this manual.

At the time of writing this manual, the Color Scheme and Screen Resolution functions have not yet been enabled. It is likely that some of the functionality will be modified and/or enhanced from the following description. Please see release notes in future QDT releases and watch for updates to this manual for up to date information on these functions.

Information Displayed

Background Color

In the Display Background section, the selected background color (regardless of Background Type selection) is displayed along with its 24 bit real color value in hexadecimal.

Startup Resolution

This is the screen resolution that this session started in.

Current Resolution

This is the currently displayed screen resolution.

User Entries

Background Type – None, Pure Color, Image

{ exclusive list }

When QDT starts up or the 'Apply' button in this section is pressed, the display's background will be updated if either 'Pure Color' or 'Image' has been selected. Pure color will use the color shown in the box to the right of this menu, while Image will use the file given in the text entry box just below the list.

No updates will happen by just picking a different entry on the list. However, if the desktop is saved, this change will occur in future QDT startups.

Color

{ button – program, waits for program to close }

This calls the QPC color picker helper program which displays a color palette plus favorite colors. By selecting a color in this program and exiting it, the color displayed in the box next to the button will be updated. This color change will be applied at the next QDT startup or if this section's 'Apply' button is pushed and the Background Type is set to Pure Color.

See the appendix section for 'QCP usage notes' for specific details on how to use the QCP color picker application.

Image

{ text entry box – max length = 25 chars }

This entry refers to a _SCR file that is applied to the background if the Background Type is set to Image and either QDT is restarted or the 'Apply' button in this section is pressed.

SCR format files must be located in the BACKGNDS directory located in the location pointed to by the QPCICON_LOC_ environmental variable.

CAUTION: *The correct size and format must be used or else the results are undefined and may potentially cause system instability or crashes. It is up to the user to ensure this.*

There are three common formats used on QPC capable systems.

Mode 16 : Aurora 8 bit color mode
Mode 32 : QCP 16 bit color mode
Mode 33 : Q40/Q60 16 bit color mode

These are not interchangeable between systems with the exception that QPC can be set to start in Aurora Mode 16.

The default supplied backgrounds use a naming scheme (optional) where the resolution is part of the name to allow for easier choosing. For example, one supplied background is:

qdtb1280x1024

This is a 'qdt' default, 'b' blue background for a 1280x1024 pixel sized screen.

The LMB as usual allows for direct text input of the file name. Do not include the _SCR suffix here (it must be part of the actual file name though) and do not include the directory path.

The RMB opens a list window showing all _SCR files found in the proper location, from which the user can choose.

Display Background : Apply

{ button – action }

As mentioned, since display backgrounds are a critical change, this separate 'Apply' button must be picked to cause the screen background to update, depending on what settings are in this section.

After an update and the notebook is closed, all the desktop icons will automatically refresh themselves so that their transparent parts inherit the new background.

Color Scheme – QDT Default, Sys Palette, Theme

{ exclusive list }

In QDT, the color scheme is made up of four parts; 1) the colors used in all the QDT windows, 2) the desktop background image or color, 3) the folder default background image or color, and 4) the icons.

The initial scheme is the QDT default set. The two other choices available are the System Palette which is only a color change and does not affect the backgrounds or icons, and the Theme. Themes are packaged combinations that can affect all the elements mentioned in the prior definition of a color scheme. { at the time of this manual preparation, only the QDT Default method is active }.

The theme will not be updated until the next restart of QDT (assuming the desktop is saved) or the 'Apply' button in this section is pressed.

Theme

{ text entry box – max length = 64 chars }

Theme files have the suffix of *THM* and must be located in *THEMES* directory within the primary QDT directory (QDT_LOC).

The LMB allows direct text entry. Do not include the directory or the file suffix.

The RMB or shortcut key brings up a list window of *_THM* type files available in the proper location.

System Pal – 0 to 3

{ scrolling list }

SMSQ/E has 4 system palettes available. It is possible to use any of these palettes for the QDT colors used in its windows. Set the number here by either clicking on the arrows or using the 'Up' or 'Down' keys on the keyboard if the user has clicked into the box with the number.

Color Scheme : Apply

{ button – action }

By pressing this button, whatever Color Scheme method chosen is applied directly on the current desktop.

Resolution Change to:

{ exclusive list }

This list displays a series of standard resolutions, allowing up to the maximum of resolution equal to the startup resolution displayed just above the list. It should be noted that all systems can not support these resolutions.

For other resolutions than what are shown here, the Custom choice is available. If the Custom option is picked, then both X and Y values must be entered to the right of this list before applying the change. If a resolution larger than the startup resolution is chosen, it will be adjusted down to match the startup value when applying.

The change will not occur until the 'Apply' button in this section is specifically pressed.

CAUTION: when applying a smaller resolution that is currently displayed, make sure all windows fit within this resolution. Icons outside of the area will be redrawn along the edge and may overlap each other. {we are looking at a feature that will automatically adjust icon placement}

Custom Values : X:

{ text entry box – max length = 4 chars }

This is the horizontal resolution in pixels of the custom screen size that is desired. It must be numbers only and can not be larger than the startup values.

Custom Values : Y:

{ text entry box – max length = 4 chars }

This is the vertical resolution in pixels of the custom screen size that is desired. It must be numbers only and can not be larger than the startup values.

Screen Resolution : Apply

{ button – action }

By pressing this button, the screen resolution will change to the values set in this section. If the system does not support the chosen resolution, in most cases no changes will occur. This is very system dependent.

ScreenSaver

Tight integration for two different screen savers is scheduled to be included; Cuedark and Dilwyn Jones' screen saver. At this time the implementation is not included and all controls in this window will have no effect on the system.

Since this page will be changing, this manual edition will not cover the specific entries.

QDOptions

This page includes QDT specific options that were initially setup during the install process but are user configurable.

Information Displayed

-none-

User Entries

Text Viewer

{ text edit box – max length = 64 chars }

This is the program that will, by default, be used to view any files that are of type 'text'. If another program or association is tied to an individual file object, then that will override this default behavior.

The selected program must be able to take the file's name to be viewed/edited as a direct command line. For example, the QD editor can be started with:

QD;"win1_boot"

This will load the file win1_boot directly into QD when it opens.

This change is effective immediately.

LMB allows direct text entry which should include the full path and program name.

RMB or the hot key brings up a QMenu file/directory to select the file to use.

Graphics Viewer

{ text edit box – max length = 64 chars }

This is the program that will, by default, be used to view any files that are of type 'graphic'. The same rules apply here as do for the Text Viewer discussed previously.

BaseColours Location

{ text edit box – max length = 64 chars }

The QCP color picker program used for background colors and in the Theme Manager uses a file called BaseColours_inf which holds the user's favorite colors (see the QCP usage appendix for more information). By default, a copy of this file is kept in the normal QDT_BIN_ directory.

However, if a user is already using this program and has an existing BaseColours_Inf in a different location, then this entry can redirect the QDT version of QCP to use the user's original location. This will allow using the same 'favorite' colors whether the program is called from QDT or run independently.

This entry is for the directory that holds the original BaseColours_inf file and should not include the actual file name. As normal, the LMB allows direct text entry.

The RMB or shortcut key brings up a QMenu Directory selection window for selecting from.

Hot Key

{ text edit box – max length = 1 char }

This is where the user can change the QDT hotkey that calls up the QDT drop-down menu from just about anywhere. The entry must be a single letter {Function keys are not yet supported in QDT}.

Hot Key : Apply

{ button – action }

Pressing the 'Apply' button will change the hotkey to the requested one, written into the user's boot file. The change itself will not take effect until the QL system itself is rebooted {at this time – an update to QDT will offer an immediate response without rebooting}.

If the selected key is already in use by the hotkey system for some other application, then a warning message will allow the user to change their mind and abort, or to go ahead and assign it as requested.

SysOptions

- not implemented yet –

This section will offer system based options for the user to reconfigure.

MenuItems

- not implemented yet –

This section will allow customization to the QDT drop down menus.

HardwareInfo

- not implemented yet –

This section will supply information on the hardware side of the user's system.

SysSoftwareInfo

- not implemented yet-

This section will supply information on software running on the user's system.

Executable File

This notebook is opened from any executable program's icon drop-down menu – Properties.

Object

All text entry boxes on this page have standard LMB, RMB and hotkey access unless specifically stated.

Information Displayed

Config Block Name

This is the program name found in the standard QJump type config block. Not all programs have Config blocks and some do not support the name feature. In these cases either a message (block not found) or possibly nonsense text will be displayed.

Config Block Vers

This is the program's version number also found in the QJump type config block.

Object Type

This is the QDT object type. In this case it will display Executable File.

User Entries

Object Name

{ text entry box – max length = 31 chars }

This is the name of the executable program itself. It should include any suffixes but not the directory.

If the RMB or hot key is used, after a file is selected from the QMenu file/directory window, then both this Object Name and the Object Location will be updated.

Object Location

{ text entry box – max length = 64 chars }

This is the directory that the executable file is physically located in.

Data Dir

{ text entry box – max length = 64 chars }

When the executable file is run, this directory will be set as the Data Directory that the program will use. If the 'sd' symbol is shown, it will use the displayed value taken from the QL 'data default'.

Command Line

{ text entry box – max length = 64 chars }

This is the command line passed to the executable file when it is run. Spaces are OK within this line. For quotes, use single quotes only.

For example, if Command Line = -l win1 -r 'garbage entry', then the following is done to launch this program (assuming a program directory/name = win1_test_exe):

```
EX win1_test_exe;"-l win1 -r 'garbage entry'"
```

Only text entry is allowed for this item so RMB and hotkey have the same affect as LMB does.

Program Dir

{ text entry box – max length = 64 chars }

When the executable file is run, this directory will be set as the Program Directory that the program will use. If the 'sd' symbol is shown, it will use the displayed value taken from the QL 'program default'.

Output Dev

{ text entry box – max length = 44 chars }

This is the default output device that the executable program will use. If the 'sd' symbol is shown, the program will use the displayed value taken from the QL 'output device' value.

Session

At the time of this manual, this page has not yet been implemented. Parts of this page are likely to change.

Information Displayed

Executable Type

This will specify if the file is standard executable, ProWesS, or other type of application.

Language

This will likely not be used.

User Entries

Object Open Behavior : When Opening

When an executable file is opened, if the 'Display existing window' is selected, QDT will check to see if it has already launched the program. If it has, it will pick the program to the top. If it hasn't launched the program it will launch a new version of it.

QDT will ignore any copies that it did not launch.

If 'Create new window' is selected, then QDT will always start a new version of the program, regardless of any other copies currently running.

Operating Mode

This option mirrors the QPtr ways of launching a program.

Add pointer OUTLN

If selected, a QPtr type outline will be put around the launched program so that it can be seen by the pointer environment.

Associations

At the time of this manual, this page has not yet been implemented.

In addition to the default viewers that QDT uses, QDT both adds its own association capabilities and integrates into the FileInfo capabilities.

Information Displayed

-none-

User Entries

Associated File Type

This is the file type to be associated with this program for use within the QDT environment.

Associated File Suffix

This is the suffix on a file that will be associated with this program as used within the QDT environment.

Set to Default : QDT

If a default value exists within QDT, this resets the Type and Suffix to that default value.

Set to Default : FileInfo

If the FileInfo system has a default value for this program, this resets the Type and Suffix to the FileInfo default value.

ConfigBlkProp

At the time of this manual, this page has not yet been implemented.

This page will show the program's Config block variables and settings if a Config block is found and allow the user to change them directly.

IndFileProp

At the time of this manual, this page has not yet been implemented.

Certain specific programs hold their configuration variables in separate files. The more common ones will be supported on this page.

Folder – Normal and Executable

This notebook is opened from any executable or standard folder's icon drop-down menu – Properties selection. If there is a difference between a normal or executable folder's page entry, the individual item will be marked [normal] or [exec].

Object

All text entry boxes on this page have standard LMB, RMB and hotkey access unless otherwise specified.

Information Displayed**Config Block Name [exec]**

This is the program name found in the standard QJump type config block. Not all programs have Config blocks and some do not support the name feature. In these cases either a message (block not found) or possibly nonsense text will be displayed.

Config Block Vers [normal]

This is the program's version number also found in the QJump type config block.

Object Type

This is the QDT object type. It can for this type of object display either Folder or Executable Folder.

User Entries**Object Name**

{ text entry box – max length =32 chars }

This is the name that is used in the open folder's title section. Only direct text entry is available for this item.

Executable File [exec]

{ text entry box – max length = 64 chars }

If an executable folder is executed (default behavior) instead of opened, this is the file name that is executed. It should include any file suffixes but not the directory that the file is in.

If the RMB or hotkey is used and a selection is made with the QMenu window, then both this entry and the Executable Location entries will be filled in.

Executable Location [exec]

{ text entry box – max length = 64 chars }

This is the location of the executable file for this folder object.

Change

{ button – action }

This will switch the object back and forth between being a normal folder and executable folder.

Data Dir

{ text entry box – max length = 64 chars }

When the executable file runs, this directory will be set as the Data Directory that the program will use. If the 'sd' symbol is shown, it will use the displayed value taken from the QL 'data default'.

Command Line [exec]

{ text entry box – max length = 64 chars }

This is the command line passed to the executable file when it is run. Spaces are OK within this line. For quotes, use the single quotes only.

For example, if Command Line = -l win1 -r 'garbage entry', then the following is done to launch this program (assuming a program directory/name = win1_test_exe):

```
EX win1_test_exe;"-l win1 -r 'garbage entry'"
```

Only text entry is allowed for this item so RMB and hotkey have the same affect as LMB does.

Program Dir

{ text entry box – max length = 64 chars }

When the executable file is run, this directory will be set as the Program Directory that the program will use. If the 'sd' symbol is shown, it will use the displayed value taken from the QL 'program default'.

Output Dev

{ text entry box – max length = 44 chars }

This is the default output device that the executable program will use. If the 'sd' symbol is shown, the program will use the displayed value taken from the QL 'output device' value.

Session [exec only]

At the time of this manual, this page has not yet been implemented. Parts of this page are likely to change.

Information Displayed**Executable Type**

This will specify if the file is standard executable, ProWesS, or other type of application.

Language

This will likely not be used.

User Entries**Object Open Behavior : When Opening**

When an executable file is opened, if the 'Display existing window' is selected, QDT will check to see if it has already launched the program. If it has, it will pick the program to the top. If it hasn't launched the program it will launch a new version of it.

QDT will ignore any copies that it did not launch.

If 'Create new window' is selected, then QDT will always start a new version of the program, regardless of any other copies currently running.

Operating Mode

This option mirrors the QPtr ways of launching a program.

Add pointer OUTLN

If selected, a QPtr type outline will be put around the launched program so that it can be seen by the pointer environment.

Associations [exec only]

At the time of this manual, this page has not yet been implemented.

In addition to the default viewers that QDT uses, QDT both adds its own association capabilities and integrates into the FileInfo capabilities.

Information Displayed

-none-

User Entries

Associated File Type

This is the file type to be associated with this program for use within the QDT environment.

Associated File Suffix

This is the suffix on a file that will be associated with this program as used within the QDT environment.

Set to Default : QDT

If a default value exists within QDT, this resets the Type and Suffix to that default value.

Set to Default : FileInfo

If the FileInfo system has a default value for this program, this resets the Type and Suffix to the FileInfo default value.

MenuItems

- not implemented yet –

This section will allow customization to the QDT drop down menus.

ConfigBlkProp [exec only]

At the time of this manual, this page has not yet been implemented.

This page will show the program's Config block variables if a Config block is found and settings and allow the user to change them directly.

IndFileProp [exec only]

At the time of this manual, this page has not yet been implemented.

Certain specific programs hold their configuration variables in separate files. The more common ones will be supported on this page.

File – Text, Graphics, and Unknown

This notebook is opened from any file's icon drop-down menu – Properties.

Object

Information Displayed

Config Block Name

This is the program name found in the standard QJump type config block. Not all programs have Config blocks and some do not support the name feature. In these cases either a message (block not found) or possibly nonsense text will be displayed.

Config Block Vers

This is the program's version number also found in the QJump type config block.

Object Type

This is the QDT object type. It will display in this case either a Text or Graphics or Other File.

User Entries

Object Name

{ text entry box – max length = 31 chars }

This is the name of the executable program itself. It should include any suffixes but not the directory.

If the RMB or hot key is used, after a file is selected from the QMenu file/directory window, then both this Object Name and the Object Location will be updated.

Object Location

{ text entry box – max length = 64 chars }

This is the directory that the executable file is physically located in.

Change Object Type – Text, Graphics, Other

{ exclusive list }

This allows the change between object types. Primarily this will affect which default viewer will be used if no other specific viewer/program is specified for this file.

Session

At the time of this manual, this page has not yet been implemented.

Information Displayed

-none-

User Entries

Object Open Behavior : When Opening

When an executable file is opened, if the 'Display existing window' is selected, QDT will check to see if it has already launched the program. If it has, it will pick the program to the top. If it hasn't launched the program it will launch a new version of it.

QDT will ignore any copies that it did not launch.

If 'Create new window' is selected, then QDT will always start a new version of the program, regardless of any other copies currently running.

Associations

At the time of this manual, most of this page has not yet been implemented. It is likely that this page will change.

Information Displayed

-none-

User Entries

File Type – Text, Graphics, Other

{ exclusive list }

This allows the change between object types. Primarily this will affect which default viewer will be used if no other specific viewer/program is specified for this file. This has the same function as does Change Object Type on the Object page.

Run with program

{ text entry box – max length = 64 }

If this is set to an executable file, it will override any other association for this specific object. The entry should be the full file and directory, including suffixes.

The selected program must be able to take the file's name to be viewed/edited as a direct command line. For example, the QD editor can be started with:

QD;"win1_boot"

As usual, the RMB or shortcut key will bring up a QMenu file/directory window for selection.

Printer

This object will not be implemented.

File Manager

This object is being planned but has not been implemented.

Thing

This object will not be implemented.

Appendixes

A: Software License and Warranty

The following is a direct copy of the license that is included with the QDT package. It can be read from within the QDT installer where the user agreement is required before the installation can be continued. The text can also be found in the QDT DOCS_ directory as "license_txt".

End-User License Agreement for QDT (QL Desktop)
All Official Full Release Versions

Copyright © 2004-2014 James D Hunkins
JDH Software Technologies
➡ All rights reserved.

This End-User License Agreement (EULA) is a legal agreement between you (either an individual or a single entity) and the author (James Hunkins - JDH Software Technologies) of this software product (QDT), which includes computer software and associated media ("SOFTWARE PRODUCT").

By installing, copying, or otherwise using the SOFTWARE PRODUCT, you agree to be bounded by the terms of this EULA. If you do not agree to the terms of this EULA, do not install or use the SOFTWARE PRODUCT and destroy all copies in your possession.

SOFTWARE PRODUCT LICENSE

QDT is being distributed as Freeware for personal or commercial use. It may be included with other software distributions. You are NOT allowed to charge for distributing this software other than for the material and shipping costs (without markup).

1. GRANT OF LICENSE

This EULA grants you the following rights:

Installation and Use: You may install and use an unlimited number of copies of this SOFTWARE PRODUCT.

Reproduction and Distribution: You may reproduce and distribute an unlimited number of copies of the SOFTWARE PRODUCT; provided that each copy shall be an un-altered and complete copy, including all copyright and trademark notices, and shall be accompanied by a copy of this EULA.

The SOFTWARE PRODUCT may be included in any free or non-profit packages or products.

2. DESCRIPTION OF OTHER RIGHTS AND LIMITATIONS

You may not reverse engineer, decompile, or disassemble the SOFTWARE PRODUCT or change (add, delete or modify) the resources in the compiled code without express written permission of the copyright owner.

Software Transfer: You may permanently transfer all of your rights under this EULA, provided the recipient agrees to the terms of this EULA.

Termination: Without prejudice to any other rights, the Author of this Software may terminate this EULA if you fail to comply with the terms and conditions of this EULA. In such event, you must destroy all copies of the SOFTWARE PRODUCT and all of its component parts.

3. COPYRIGHT.

This SOFTWARE PRODUCT is owned by JDH Software Technologies (James Hunkins) and is protected by United States and Canadian copyright laws and international treaty provisions. Therefore, you must treat the SOFTWARE PRODUCT like any other copyrighted material (e.g., a book or musical recording). Some of the included supplemental software programs are supplied other other authors/owners who hold the respective copyrights as noted in the package documentation and must be treated in the same copyrighted material manner.

4. LIMITED WARRANTY

NO WARRANTIES.

The Author of this Software expressly disclaims any warranty for the SOFTWARE PRODUCT. The SOFTWARE PRODUCT and any related documentation is provided "as is" without warranty of any kind, either express or implied, including, without limitation, the implied warranties or merchantability, fitness for a particular purpose, or non-infringement. The entire risk arising out of use or performance of the SOFTWARE PRODUCT remains with you.

NO LIABILITY FOR DAMAGES.

In no event shall the author of this Software be liable for any special, consequential, incidental or indirect damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or any other pecuniary loss) arising out of the use of or inability to use this product, even if the Author of this Software is aware of the possibility of such damages and known defects.

This Agreement shall be governed by the laws of the province of Ontario, Canada. If for any reason a court of competent jurisdiction finds any provision of the Agreement, or portion thereof, to be unenforceable, that provision of the Agreement shall be enforced to the maximum extent permissible so as to effect the intent of the parties, and the remainder of this Agreement shall continue in full force and effect.

Support and Additional Material Supplied By:

Graphics/Color Development Support:
Thorsten Herbert, Italy

Additional Icons By:
Roy Wood, England

Included Non-licensed code : no charge distribution

FileInfo2

FI2config_obj

FileInfo2_bin

FileInfo2_txt

FI2config_help

Copyright © 1993-1999 Thierry Godefroy

MenuConfig 02

Copyright © 1992-2003 Jochen Merz Software

Recommended additional software (not included):

QMenu menu extensions

Copyright© 1989-97 Jochen Merz Software

B: Notebook Entries : by name and page

- Desktop : Object
 - Output Device
- Desktop : IconOpen
 - Icon Text
 - Icon Name
 - Icon Size – Small, Normal
 - Edit** Icon
 - IconDraw**
 - Find** Icon
- Desktop : Startup
 - Desktop Startup File
 - Show Desktop Object
 - Alternate File
 - Change** Alternate File
 - Restart** (and change) Alternate File
 - Save **Current** Desktop
 - Save **Other** Desktop
 - Theme** Manager
- Desktop : Archive
 - Archive Desktop
 - At Every QDT startup
 - At next QDT startup only
 - If Desktop has Changed
 - Per number of Days
 - If Manually Saving Desktop
 - Show Archive status during startup
 - Keep n separate archives
 - Archive Location
 - Recover Archived Desktop – list
 - Recover** desktop
- Desktop : Display
 - Background
 - Type – None, Pure Color, Image
 - Image
 - Color**
 - Apply** Background
 - Color Scheme
 - List – QDT Default, Sys Palette, Theme
 - Theme
 - System Pal #
 - Apply** Color Scheme
 - Screen Resolution
 - Change to (list)
 - Custom Values X
 - Custom Values Y
 - Apply** Screen Resolution
- Desktop : ScreenSaver
 - Program Directory/Name
 - Module Directory
 - Module Suffix
 - Current Module
 - Enable
 - Config**
- Desktop : QDOptions
 - Text Viewer
 - Graphics Viewer
 - BaseColours Loc

Hot Key
Apply HotKey
 Desktop : SysOptions
 Desktop : MenuItem
 Desktop : GlobalVar
 Desktop : HardwareInfo
 Desktop : SysSoftwareInf

 Exec File : Object
 Object Name
 Object Locations
 Data Directory
 Command Line
 Program Directory
 Output Device
 Exec File : Session
 Object Open Behavior : When Opening
 Pointer Env Behavior : Operating Mode
 Add pointer OUTLN
 Exec File : IconClosed
 Icon Text
 Icon Name
 Icon Size – Small, Normal
 Edit Icon
 IconDraw
 Find Icon
 Exec File : IconExec
 Icon Text
 Icon Name
 Icon Size – Small, Normal
 Edit Icon
 IconDraw
 Find Icon
 Exec File : Associations
 Associated File Type
 Associated File Suffix
 Set to Default : **QDT**
 Set to Default : **FileInfo**
 Exec File : ConfigBlkProp
 Exec File : IndFileProp
 Exec File : GlobalVar

 Folder-Normal : Object
 Object name
 Change to executable
 Data Directory
 Program Directory
 Output Device
 Folder-Normal : IconClosed
 Icon Text
 Icon Name
 Icon Size – Small, Normal
 Edit Icon
 IconDraw
 Find Icon
 Folder-Normal : IconOpen
 Icon Text
 Icon Name

- Icon Size – Small, Normal
- Edit** Icon
- IconDraw**
- Find** Icon
- Folder-Normal : MenuItems
- Folder-Normal : GlobalVar
- Folder-Exec : Object
 - Object Name
 - Executable File
 - Executable Location
 - Change** to normal
 - Data Directory
 - Command Line
 - Program Directory
 - Output Device
- Folder-Exec : Session
 - Object Open Behavior : When Opening
 - Pointer Env Behavior : Operating Mode
 - Add pointer OUTLN
- Folder-Exec : IconClosed
 - Icon Text
 - Icon Name
 - Icon Size – Small, Normal
 - Edit** Icon
 - IconDraw**
 - Find** Icon
- Folder-Exec : IconOpen
 - Icon Text
 - Icon Name
 - Icon Size – Small, Normal
 - Edit** Icon
 - IconDraw**
 - Find** Icon
- Folder-Exec : Associations
 - Associated File Type
 - Associated File Suffix
 - Set to Default : **QDT**
 - Set to Default : **FileInfo**
- Folder-Exec : MenuItems
- Folder-Exec : ConfigBlkProp
- Folder-Exec : IndFileProp
- Folder-Exec : GlobalVar
- File : Object
 - Object Name
 - Object Location
 - Change Object Type – Text, Graphics, Other
- File : Session
 - Object Open Behavior : When Opening
- File : IconClosed
 - Icon Text
 - Icon Name
 - Icon Size – Small, Normal
 - Edit** Icon
 - IconDraw**
 - Find** Icon
- File : IconOpen

Icon Text
Icon Name
Icon Size – Small, Normal
Edit Icon
IconDraw
Find Icon
File : Associations
Associated with Executable Program – list
File Type – text, graphics, other
SubFile Type
Run with program
Change all objects; make default
FileInfo program
File : Global Variables

C: QCP Usage Notes

QCP (QL ColourPicker) Version 1,1

© Wolfgang Uhlig 2004

Integrated HSVprocs are © Marcel Kilgus 2003

minor bug fixes and adaptation to QDT by J.D.Hunkins

QCP allows easy picking of any color along with saving of up to 16 'favorite' colors. QDT integrates QCP and uses it when picking a color for the Desktop background (Desktop Notebook) and in the theme manager for different color selections.

To pick a color, simply click on the color of choice in the larger window with the full color spectrum. Then, to choose the deepness of the color, either pick directly in the vertical window next to the larger window or move the arrow to the preferred position. The final chosen color is displayed dynamically in the smaller square window just to the left of the QCP text. The numerical representation of this color is displayed just below this window in the TrueColour and SystemColour windows.

The 16 rectangles along the bottom of the screen are favorite colors that the user can change and save for next time. To change any of the colors, click first on the one to change. Then pick a new color as just described. When ready, click on 'add' and the chosen rectangle will update with the new color and will be saved when exiting the program if desired.

To exit the program and transfer the color back to QDT, there are three options: 1) click on 'send', 2) click on the red right pointing arrow loose item, 3) or simply hit escape. Note that this behavior is slightly different than when running QCP directly versus from within QDT.

Loose items are standard QL types. From left to right, they are Info, move, sleep, and close. In most cases, when running from QDT, QDT will lock the window which calls this program. So if the user puts QCP to sleep (IE: not actually closing it), the original QDT window will remain locked. The user must close QCP before continuing with what they were doing in QDT.

basecolours_inf

If the user is already using a copy of QCP or any of the other programs that use basecolours_inf to store favorite colors, they can direct QDT to use the basecolours_inf file that they have been previously using instead of the default one that came with QDT. In the Desktop Notebook, QDToptions tab, the BaseColours Loc: can be set to the location of the original file. QCP from within QDT will then show the same favorite colors as external use of QCP and other programs will show.

D: Capturing a Background Image for use in QDT

QDT uses the SMSQ/E BGIMAGE capability to place a background image. This image must be in the standard QL _SCR format (IE: a direct capture of the display when the image is on the screen). It is resolution and color mode dependent. A screen captured from a QPC in mode 32 will not work on a Q40 or on an Aurora card, for example.

There are different ways that an image can be prepared as a background to be used in QDT. Two options are presented here.

Method 1 : converting to a QL sprite on a Windows system

- 1) Windows machine : generate a PNG file of the image that is the same size as the QL screen
- 2) Windows machine : run PNGConv_exe (Marcel Kilgus' PNG to sprite conversion program)
 - set to 8 bit : mode 16 : for Aurora
 - set to 16 bit : mode 32 : for QDT or Q40/Q60-> generate a binary file
- 3) move the binary _SPR file to the QL system that will be displaying the background
 - do not work on an Aurora card and expect it to display on a Q40 for example
- 4) exec snatch4_obj (Dilwyn Jones' screen grabber program)
 - set the final destination and name of the desired _SCR file
 - tell it to go (after noting the shortcut key to capture the screen)
- 5) load and run the following SBasic program to load the sprite to the desktop
 - this program is a very simple program and the system will need to be reset when done
 - it allows the user to capture multiple screens in one session, just enter the largest size _SPR file to start with

NOTE: *this program is not guaranteed in any way but is supplied to help give an idea on how to do this; use at your own **RISK!***

```
100 INPUT #0,"Sprite file size: ";size
110 nr = RESPR(size + 10): REMark added 10 just for safety
120 INPUT #0,"SpriteName [_spr added] [hit enter if done]: ";name$
130 IF name$ == "": STOP
135 LBYTES name$ & "_spr",nr
140 INPUT #0,"Screen width: ";screenx
150 INPUT #0,"Screen height: ";screeny
160 WINDOW #1, screenx, screeny, 0, 0
165 CLS #1
170 WSPRT #1,0,0,nr
180 REPEAT getchar: z$=INKEY$(-1): IF z$ = " ": EXIT getchar
190 WMON ,0,42
210 GO TO 120
```

The above files can be downloaded from:

PNGConv_exe	http://www.kilgus.net
snatch4_obj	http://www.dilwyn.me.uk/indexfrm.html

Method 2 : converting from a BMP format on a QL system

- uses WL_BMPCVT32 (mode 32, QDT) or WL_BMPCVT33 (mode 33, Q40, Q60) SBasic keywords to do the conversion.

- these are included in Wolfgang Lenerz's BMPCVT_bin extensions

- 1) Copy a BMP 24 bit image (Windows format) to be copied to a QL system
 - needs to be the pixel size of the QL screen to be displayed on
- 2) make sure that the BMPCVT_bin file has been LRESPRed (IE: the extensions are loaded)
- 3) type (using WL_BMPCVT32 or WL_BMPCVT33)
WL_BMPCVT32 "<devx>_my_PCbmp_picture",<devy>_my_QL_picture"

4) test it with:

BGIMAGE "<devy>_my_QL_picture"

There are extra options on the conversion but they either crop the image or add a border. So it is best if the correct BMP image size can be started with.

BMPCVT_bin can be gotten from <http://www.dilwyn.me.uk/indexfrm.html>
- look in the graphics section