Apple IIGS
#24: *Apple IIGS Toolbox Reference* Updates

Revised by: Dave Lyons — May 1992
Written by: Rilla Reynolds, Matt Deatherage, Dave Lyons, C.K. Haun & Eric Soldan — October 1987

This Technical Note documents changes to the *Apple IIGS Toolbox Reference* manuals. Please contact Apple II Developer Technical Support at the address listed in Apple II Technical Note #0 if you have additional corrections or suggestions for any of the Apple IIGS Toolbox documentation.
Changes since December 1991: Added corrections to Dialog Manager, Menu Manager, Tool Locator, Window Manager, and Appendix E.

The current Apple IIGS Toolbox reference material is *Apple IIGS Toolbox Reference*, volumes 1 to 3 as well as this Technical Note. (The *Apple IIGS Toolbox Reference Update* beta draft from APDA is obsolete and should **not** be used.)

Corrections to Volume 1

Desk Manager—FixAppleMenu Can Die With Error $0512

Fatal system error $0512 comes from `FixAppleMenu` (in the Desk Manager). It means that one of your installed New Desk Accessories does not have a well-formed menu title string. In particular, the required backslash (\) character was not found (make sure bit seven is off).

Dialog Manager—`editLine` Item Value

On page 6-12, the description of an `editLine` item value should read "Maximum length of the item text (0 to 255 characters)."

The List Manager Wants the Port Set Properly

The List Manager expects the current `grafPort` to be set properly before you make most List Manager calls; drawing can occur in funny places if the `grafPort` is not set properly before calls that draw (like `SelectMember2`). Most List Manager calls, and many other toolbox calls, require that the current `grafPort` be explicitly set. Before you call List Manager routines that draw, set the current port to your window with a `SetPort` call. Remember the note in Volume 2 under the `NewWindow` call—"Important: `NewWindow` does not set the current port, but many routines require that a current port exist. Use the QuickDraw II routine

`SetPort` to set the current port." Using `SetPort` can prevent toolbox confusion and reduce your debugging time.

DeleteMItem Operates on the Current Menu Bar

Page 13-37 says `DeleteMItem` removes the specified item from the current menu. It means the item is removed from the current menu bar.

Error $0F02 from GetMItem

`GetMItem` returns error $0F02 if the specified menu item is not found.

On page 13-45, the return value from `GetMenuFlag` should read "Word—`menuFlag` value for the specified menu."

On page 13-56, in the description of the `hiliteFlag` parameter to `HiliteMenu`, no particular value of "TRUE" is specified. $0001 is a good value ($8000 does not work; bit 15 is special).

On page 13-72, `SetMenuFlag` doesn't bother to actually explain what it does. If bit 15 of `newValue` is zero, each set bit set forces the corresponding bit in the menu's flag value to be set. If bit 15 of `newValue` is one, each clear bit forces the corresponding bit in the menu's flag value to be clear. Knowing this, you can set or clear more than one bit at a time, if you want.

SetVector Reference Numbers

On page 14-62, vector reference number $002C is listed as "Message pointer vector." $002C is actually the stack-based GS/OS call vector. (The real message pointer vector is not accessible through `GetVector` and `SetVector`.)

Getting a clean Mouse Mode from `ReadMouse`

On ROM 3 computers, the mouse mode byte returned from `ReadMouse` sometimes has extra bits set in the high nibble. Before feeding a `ReadMouse` value to `SetMouse`, mask off all but the low nibble (`AND #$000F`).

ReadAsciiTime Result Buffer

The description of `ReadAsciiTime` (in the Miscellaneous Tools) on page 14-16 should say the most significant bit (not byte) of each character is set to one.

SystemEvent Is All Backwards

Although applications still should not call `SystemEvent`, we should note for completeness that the input parameters listed in Volume 1 are exactly backwards in the stack diagram.

Corrections to Volume 2

QuickDraw Auxiliary Error Codes

Following are some error codes from QuickDraw Auxiliary that are not listed in volume 2.

```
$1210: picEmpty
$1211: picAlreadyOpen
$1212: pictureError

$1221: badRect
$1222: badMode
```

FrameRgn Does Not Contribute to an Open Region

The description of the `FrameRgn` routine on page 16-105 in the *Apple IIGS Toolbox Reference*, Volume 2 states that `FrameRgn` will contribute to a region definition if a region is open when `FrameRgn` is called. This is **incorrect**; `FrameRgn` does not contribute to the region being defined. To add a region to another region, use `XorRgn` or `UnionRgn`.

Tool Locator, `TLMountVolume`

On page 24-21, the description of `TLMountVolume` does not bother to mention that QuickDraw II and Event Manager must be active. If they are not, you should use `TLTextMountVolume` instead.

Tool Locator, `SetTSPtr`

When using `SetTSPtr` to patch a system tool set, the Tool Locator and Desk Manager are special. See Apple IIGS Technical Note #101, Patching the Toolbox.

Window Manager, "Draw Information Bar Routine"

On page 25-23, the code to clean up the stack is incorrect. On the `sta <14`, the comment "Works because stack and direct page are equal" is no longer true—they **were** equal until the `PLY` two lines earlier. One way to correct the code is to replace `sta <14` with `sta 14,s` and `sta <12` with `sta 12,s`.

Window Manager, `InvalRect`

The description of `InvalRect` on page 25-80 claims that `InvalRect` modifies the input rectangle; the rectangle is actually not modified.

Window Manager, PinRect

On page 25-89, in the description of `PinRect`, the two greater-than comparisons should be greater-than-or-equal.

Window Manager, `SetZoomRect`

The description of `SetZoomRect` on page 25-112 refers to `fZoomed` as bit 2 in the window frame. `fZoomed` is actually bit 1, with value $0002.

Window Record Offsets

On page 25-142, note that the offsets given into the window record refer to the record as the Window Manager treats it internally, with a `wNext` field at the beginning. When dealing with a window pointer as seen by an application, you need to subtract four from the offsets shown. For example, `wPort` is $00 (not $04), and `wControls` is $C6 (not $CA).

Appendix A, "Writing Your Own Tool Sets"

At the bottom of page A-8, "`lda #$90`" should read "`lda #$8100`" for version 1.0 prototype.

On page A-10, the figure should show **two** `RTL` addresses (6 bytes) on the stack.

Corrections to Volume 3

Control Manager: Menu Events

On page 28-15, note that a Menu Event is identified by the value `wInSpecial` ($0019) in the `what` field of the task record. The menu item ID is in the low word of the `wmTaskData` field.

Control Manager: Dimmed Custom Controls

In the Draw routine for both extended and non-extended controls, the high word of `ctlParam` (which was previously undocumented) contains a flag which the definition procedure can use to draw a normal or dimmed control. The value is $0000 normally, but it is $FFFF when the control is inactive (hilite value equals $00FF), or when the control's state is tied to the window's state and the window is inactive.

Control Manager: Size Box Controls

The part code for an extended Size Box control is normally 10. If the `fCallWindowMgr` bit is set in `ctlFlag`, the part code is $80; and if the size box is managed by a Text Edit control, the part code is $84.

When a Size Box control's `fCallWindowMgr` bit is set, the control needs to pass a minimum window size to GrowWindow. It gets this value from its `ctlData` field, which you can get with `GetCtlTitle` and set with `SetCtlTitle` (the low word is the minimum height, and the high word is the minimum width). A height of zero defaults to 50, and a width of zero defaults to 130.

Desk Manager: Errors from AddToRunQ and RemoveFromRunQ

The Desk Manager chapter, page 29-6, states no errors are possible for `AddToRunQ`, but any errors from the Miscellaneous Tools routine `AddToQueue` are returned unchanged.

Page 29-8 states no errors are possible from `RemoveFromRunQ`, but any errors from `DeleteFromQueue` are returned unchanged.

Event Manager: What SetAutoKeyLimit Really Does

Page 31-6 says that `PostEvent` will add up to the new auto-key limit number of auto-key events before reverting to the rule that auto-key events are only to be posted if the event queue is empty. This is not quite right. Actually, the parameter to `SetAutoKeyLimit` is used in a size comparison on the event queue—if there are `newLimit` or more events in the queue, auto-key events will not be posted. Volume 3 incorrectly states that up to `newLimit` auto-key events will be posted; this is only true if you assume the event queue is empty before the first auto-key event comes in.

List Manager

On page 35-9, the description of `ResetMember2` does not point out an important difference between `ResetMember2` and `NextMember2`. `ResetMember2` deselects the member found, but `NextMember2` does not change the member's status.

On page 35-3, bit 5 of the `memFlag` field is defined—it makes an item inactive. To make use of this bit, you must also set bit 6 of the List control's `ctlFlag` field; if you don't set this bit, the user will still be able to select members using the mouse.

Memory Manager

If the Memory Manager detects a corrupted entry in the Out Of Memory Queue, fatal system error $0209 occurs.

Menu Manager

On page 28-65, the description of the `initialValue` field is misleading. Cross out the text "that is, its relative position within the array of items for the menu." `initialValue` is simply a menu item ID, not an offset into an array.

Page 37-7 states "Because caching does not work with menus in windows, the `InsertMenu` call automatically disabled caching for such menus." Actually, `InsertMenu` doesn't do that. You should not set the `allowCache` bit for a menu in a window.

Miscellaneous Tools: Interrupt State Record Not Always Complete

The interrupt state record returned from `GetInterruptState` (and passed to `SetInterruptState`) is not always completely filled in. The Interrupt Manager, in the interest of serving AppleTalk and serial interrupts as rapidly as possible, does not take the time to save all the items in the record until those timing-critical interrupt handlers have been called. Some items are not saved at all unless the interrupt is determined to be a `BRK` instruction. Table 1 shows all items in the current interrupt state record and when they become valid:

Record variable	When valid
`irq_A`	always
`irq_X`	always
`irq_Y`	always
`irq_S`	after serial
`irq_D`	always
`irq_P`	only on break
`irq_DB`	after serial
`irq_e`	after serial
`irq_K`	only on break
`irq_PC`	only on break
`irq_state`	after serial
`irq_shadow`	always
`irq_mslot`	after serial

Table 1—Validity of Interrupt Record

Standard File

On page 48-39, the description of `origNameRef` reads "On output, this string contains the string confirmed by the user, which may not be the same length as the default value." This sentence is confused; ignore it. The string is not changed at all; Standard File doesn't even know how long the buffer is.

Tool Locator: Notes on StartUpTools

`StartUpTools` in System Software 5.0.4 and earlier is intended to be used from applications only, not from NDAs.

The order of the `toolArray` entries in the `StartStop` record is not important. `StartUpTools` and `ShutDownTools` always start up and shut down tools in a correct order.

`StartUpTools` in System Software 5.0.4 and earlier fails to open your application's resource fork if the application's filename contains a slash (/) or if the application directory path is longer than 64 characters.

For maximum compatibility, pass your application's master user ID with any `auxID` to `StartUpTools` instead of allocating a new user ID.

Window Manager:NewWindow2 Parameters Override Template Even When You Pass NIL

The description of the `NewWindow2` call on page 52-32 is in error. The description of the `titlePtr`, `refCon`, `contentDrawPtr`, and `defProcPtr` says, “To prevent `NewWindow2` from replacing the template values, supply `NIL` pointers…” This is only true for the `titlePtr` parameter—if you pass `NIL` for any of the other parameters then the value of that parameter in your window record is also `NIL`, no matter what the template value was. In other words, if you have the value $99 stored in your template `refCon` field, and you pass `NIL` for the `refCon` value in a `NewWindow2` call, the value of the `refCon` in the returned `grafPortPtr` is `NIL`.

Appendix E: `rTextForLETextBox2` Resources

Page E-68 of Volume 3 shows a `length` field at the beginning of an `rTextForLETextBox2` resource. This field is not actually present. The length is simply the size of the resource—it is not stored redundantly.

Appendix E: rTwoRects Resources

When the two rectangles are for 320- and 640-mode, by convention the rectangle for 320 mode comes first.

Further Reference:

- *Apple IIGS Toolbox Reference*, Volumes 1–3
- Apple IIGS Technical Note #101, Patching the Toolbox